2012

Mini-Solar Power Station

Final Technical Report

Group: Brian Amann
Stephanie Erickson
Qingyu (Jace) Meng
Kaixun (Jerry) Wang

Advisors: Dr. Yuvarajan
Jeff Erickson

SD1203
NDSU
12/7/2012

Table of Contents

Project Summary

The Mini-Solar Power Station consists of two subprojects. The first subproject was to construct a satellite tracking system. This system is known as the Solar Tracker. The Solar Tracker tracks the sun at a 90° angle ensuring that a maximum amount of sunlight is collected throughout the day. The Solar Tracker contains a satellite dish covered with a reflective Mylar material, but is also convertible to allow a solar panel to be easily installed. When the Solar Tracker does not have the solar panel installed, reflective Mylar material reflects the sunlight to the dish's focal point. A hollow copper coil tube, which contains continuously running fluid, is placed at the focal point. As the sunrays are reflected and focused at the copper coil tube, essentially at one point, the copper coil will begin to heat up which will consequently heat the fluid inside the coil. Thermal insulated tubing connects the copper coil to a heat exchanger and a water reservoir. A water pump circulates water between the coil and through the heat exchanger to the water reservoir so the temperature rise from the water can be measured. The amount of heat collected from the sun can in turn be calculated. Using data acquisition, the temperature from the heat exchanger is collected and plotted in real time using LabVIEW. When the solar panel is installed, using the energy obtained a buck/boost battery charger will charge one 12-volt rechargeable battery. The battery will run a 300-watt inverter so common household appliances can be used.

The second subproject includes constructing a mechanical fixture containing a heat/light source to simulate the rising, setting, and latitudinal movements of the sun. The fixture is used in correspondence with the above-mentioned Solar Collector and serves as a testing device to prepare the Solar Tracker for operation outside. A 250W infrared heat lamp is the source of heat and is used to represent the sun. A dimmer switch is used with the heat lamp, which dims the heat lamp to represent cloud cover. As stated above, temperature data will be collected and plotted in LabVIEW. With the dimming of the lamp, the decrease in temperature should be evident in the LabVIEW plotted data. Figure 1 is the block diagram of the overall project.

Mini Solar Power Station Block Diagram

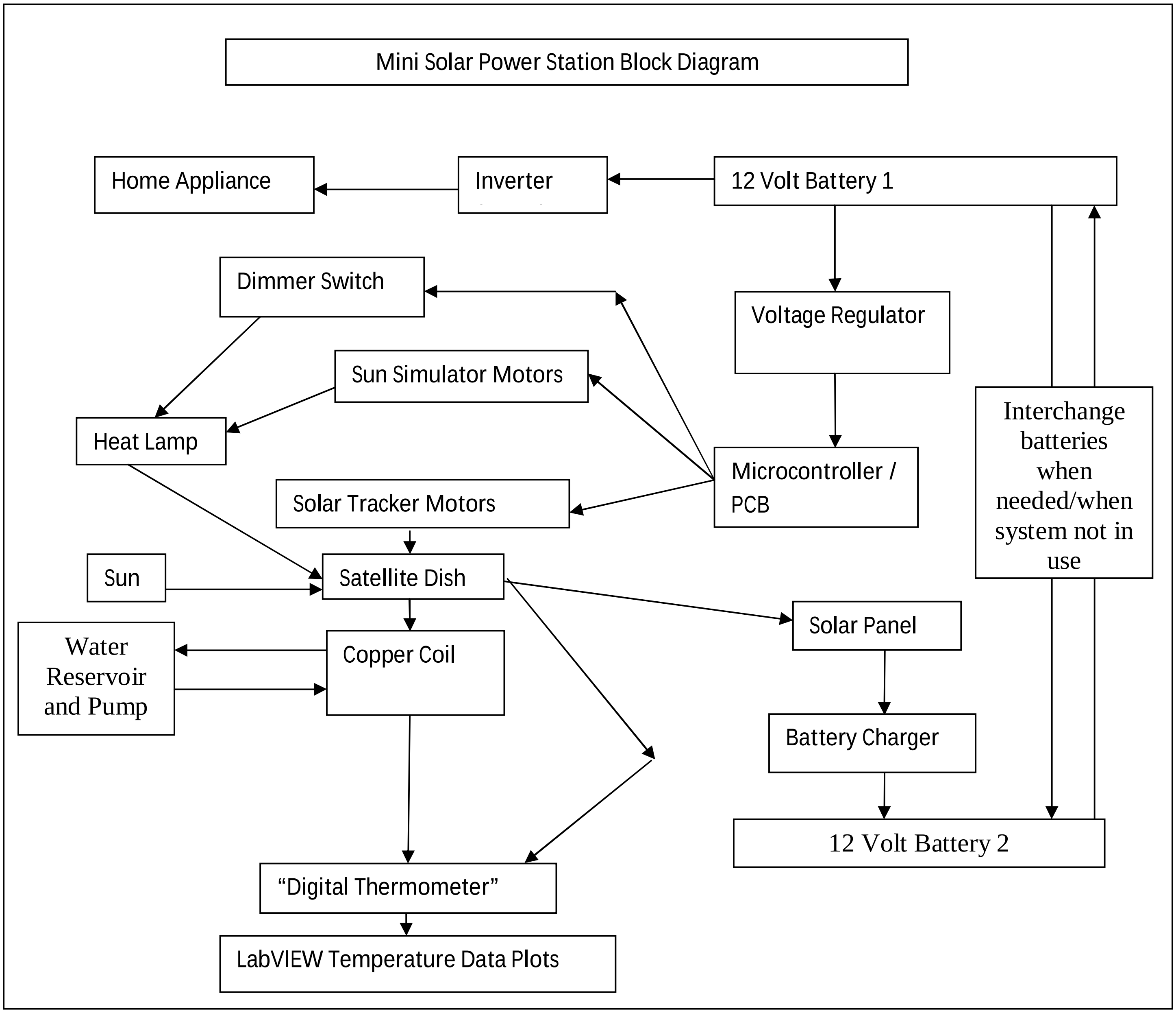


Figure 1: Mini-Solar Power Station Block Diagram

Requirements

Electric Drives

Solar Tracker

Two motors will be used to track the movement of the sun. One stepper motor will control motion of the tracker in the x direction (horizontally) and a linear actuator will control motion in the y direction (vertically).

Sun Simulator

Two motors will be used to simulate the movement of the sun. One stepper motor will move the heat bulb along the simulator track, simulating the rising and setting of the sun. A linear actuator will provide ±15 degrees of tilting in two directions for the track, simulating the latitudinal movement of the sun.

Heat and Electricity Conversion and Measurement

Solar Tracker

1. A water pump will pump water from a water reservoir through insulated tubing to the copper coil. Insulated tubing will connect the copper coil to a heat exchanger which is connected to the water reservoir on the return path, forming a closed loop system producing a continuously running water system.
2. A thermistor (heat sensitive resistor) will be placed in the heat exchanger to measure the generated heat. The heat exchanger is used to contain and measure the heat generated with the satellite system attached. As the water is heated the coil inside the box begins to heat which in turn heats the inside of the box through the process of convective heat transfer. Using a NI USB 6008 DAQ, the temperature data will be plotted in real time using LabVIEW.
3. A solar panel will be used as the energy-collecting device. Elbow clamps will be used for proper placement of solar panel onto the satellite dish. Using the energy obtained a buck/boost battery charger will charge one 12-volt rechargeable battery. A second (charged) battery will be used to run the overall system. The batteries will be interchanged when needed, preferably when the system is not in use.
4. A 300W inverter will convert the 12V DC (from the rechargeable batteries) to AC to power common household appliances.

PIC18 Microcontroller (MCU)

Solar Tracker

The MCU will receive signals from a solar sensor and command the movements of both motors accordingly. The solar sensor is built with photo resistors and a metal enclosure. This is intended to detect the direction that has the strongest sunlight.

Sun Simulator

The MCU will control both motors. Using a series of pulleys, the 10:1 geared stepper motor will be used to "push" and "pull" the sheave pulley, connected to the heat lamp, to mimic longitude sun movement. The linear actuator will be used to add ±15 degrees (from the zero starting point), thus mimicking latitude movement.

Dimmer Switch

The MCU along with pulse width modulation (PWM) will control the dimmer switch. The dimmer switch will be used with the heat lamp, which will dim and brighten the heat lamp to represent cloud cover. With the dimming of the lamp, the decrease in temperature should be evident in the LabVIEW plotted data.

Data acquisition and Monitoring

Solar Tracker

1. Using the graphical programming software LabVIEW and data acquisition (NI DAQ 6008), real time temperature of the heated water will be displayed on a graph and the current temperature will be displayed on a numerical display. There will be the option to save the temperature data for future analysis and comparison.
2. The launching of the tracking function will be controlled via a light intensity monitor. The light intensity monitor will turn the tracking function on when the sunlight intensity is above a preset level, and turn the function off when the sunlight intensity drops below the preset level. This will ensure that the system will shut off during night or when circumstances, like cloudy or rainy days, do not provide enough solar power to generate the power to activate the battery charger. The light intensity monitor will be implemented using a photo resistor, a comparator, and the MCU accompanied with proper code.

Mechanical Fixture

Solar Tracker

1. The mechanical fixture of the Solar Tracker will be composed of metal and wood. For the satellite system, a standard slim line satellite dish (one used for Direct TV satellite) and Mylar will be adhered to the inside of the satellite dish to create a reflective surface.
2. ¾" hollow copper tubing will be coiled at the satellite dish's focal point. ¾" plastic insulated tubing will be used to connect the copper coil to the water reservoir.
3. A heat exchanger will be used to measure the heat obtained. The heat exchanger will be composed of wood, cooper coil, and a fan.
4. All PCB's will be enclosed.

Sun Simulator

1. The simulator will be composed of 10' long 3/8" steel pipe, a combination of pulleys, a stepper motor, and a linear actuator to drive the heat bulb to mimic the sun's movement

2. The base structure will be made of wood.
3. All PCB's will be enclosed.

Budget

	Project Name:	Mini-Solar Power Station				
	Project Number:	SD1203				

Sun Simulator						
Mechanical Hardware	Cost/Unit	#	Purchased	Total Cost	Supplier	Notes
3/8" x 10' Steel Pipe	$21.10	2	X	$42.20	Tubes & Hose of ND	Part #: 03735
Wood (2x10-10' (1),2x4-8' (2))	$15.07	1	X	$15.07	Menards	
4 in Heavy Duty Tee Hinge	$3.87	2	X	$7.74	Grainger Industrial Supply	Part#: 1RCT3
R40 5" Dia., 250W, 120V Lamp	$21.84	1	X	$21.84	Grainger Industrial Supply	Part#: 1E295
Heat Lamp Dome	$18.00	1	X	$0.00	Free from NDSU	
Linear Actuator	$299.00	1	X	$0.00	Free from NDSU	LA34AGKS-4
Stepper Motor	$14.95	1	X	$0.00	Free from NDSU	57BYGH213
Ball Bearing Sheave	$11.91	2	X	$23.82	Grainger Industrial Supply	Part#: 5RTD5
Flat Mount Block Pulley, 2"	$7.73	2	X	$15.46	Grainger Industrial Supply	Part#: 4JX69
1/16, 50ft, 7x19 steel cable	$39.80	1	X	$39.80	Grainger Industrial Supply	Part#: 2TAY9
Solid State Relay	$30.00	1	X	$0.00	Free from NDSU	
Gear Box for stepper motor	$100.00	1	X	$0.00	Free from NDSU	Jeff Constructed
PCB	$50.00	2	X	$100.00	Advanced Circuits	Brian's & Jace's
Satellite Tracking System						
Mechanical Hardware						
Satellite Dish	$50.00	1	X	$0.00	Provided by Brian	
Stepper Motor	$14.95	1	X	$14.95	Free from NDSU	
Mylar	$11.99	1	X	$11.99	www.amazon.com	Part#: VMY130
DC Motor		1	X			
10' Copper Coil 3/8" OD	$8.74	1	X	$8.74	Menards	Item #: 675142
Brass Hose Barb	$1.99	4	X	$7.96	Menards	Model 17090298
Tetra Statuary Pump	$24.94	1	X	$24.94	Menards	
Tube to MPT connector	$2.67	4	X	$10.68	Menards	Model 17000122
PVC Insulation	$2.35	2	X	$4.78	Menards	Item #: 2KUE2
10' Plastic Tubing 3/8"	$8.49	1	X	$8.49	Menards	
12V rechargable battery	$30.00	3	X	$90.00	Free from NDSU	
NI USB 6008 DAQ	$169.00	1	X	$0.00	Free from NDSU	
Plastic Circuit Enclosures	$5.00	2	X	$20.00	Radio Shack	
350W Inverter	$29.00	1	X	$0.00	Provided by Brian	Walmart
Electrical Components						
ZTX1053a (npn transistor)	$0.98	20	X	$0.00	Free from NDSU	
MCP602 (OPAMP)	$0.62	16	X	$0.00	Free from NDSU	
PIC18F4620 (Microcontroller)	$7.92	5	X	$0.00	Free from NDSU	
ECS-200-20-46X (20 MHz Crystal)	$0.46	5	X	$0.00	Free from NDSU	
Photocell 80k-240kΩ	$1.86	15	X	$0.00	Free from NDSU	
LM7805 (5V voltage regulator)	$0.61	5	X	$0.00	Free from NDSU	
LM7812ACT (12V voltage regulator)	$0.69	2	X	$0.00	Free from NDSU	
Estimated Actual Cost of Project	$1,005.53			$468.46		

Figure 2: SD1203 Final Budget

Timeline

Design II

#	Task	Assigned To	Start	End	Dur
	Mini-Solar Power Station		1/24/12	12/7/12	229
1	Project Kick Off- Advisor Meeting #1	Brian, Stephanie, Jace, Jerry	1/24/12	1/24/12	1
2	Requirements Capture	Brian, Stephanie, Jace, Jerry	1/24/12	1/27/12	4
3	Conceptual Design, Budget, Timeline	Brian, Stephanie, Jace, Jerry	1/31/12	2/9/12	8
4	Advisor Meeting #2	Brian, Stephanie, Jace, Jerry	2/3/12	2/3/12	1
5	Mechanical Design		2/10/12	4/27/12	56
5.1	Solar Collector	Jace, Jerry	2/10/12	4/27/12	56
5.2	Sun Simulator	Brian, Stephanie	2/10/12	4/27/12	56
6	Order Electrical Components		2/10/12	2/16/12	5
6.1	Sun Simulator	Brian, Stephanie	2/10/12	2/16/12	5
6.2	Solar Collector	Jace, Jerry	2/10/12	2/16/12	5
7	Software Design		2/17/12	2/22/12	4
7.1	Sun Simulator	Brian, Stephanie	2/17/12	2/22/12	4
7.2	Solar Collector	Jace, Jerry	2/17/12	2/22/12	4
8	Software Integration and Test		2/22/12	5/3/12	52
8.1	Sun Simulator	Brian, Stephanie	2/22/12	5/3/12	52
8.2	Solar Collector	Jace, Jerry	2/22/12	5/3/12	52
9	Advisor Meeting #3	Brian, Stephanie, Jace, Jerry	2/17/12	2/17/12	1
10	Advisor Meeting #4	Brian, Stephanie, Jace, Jerry	2/24/12	2/24/12	1
11	Advisor Meeting #5	Brian, Stephanie, Jace, Jerry	3/2/12	3/2/12	1
12	Advisor Meeting #6	Brian, Stephanie, Jace, Jerry	3/9/12	3/9/12	1
13	Advisor Meeting #7	Brian, Stephanie, Jace, Jerry	3/23/12	3/23/12	1
14	Advisor Meeting #8	Brian, Stephanie, Jace, Jerry	3/30/12	3/30/12	1
15	Advisor Meeting #9	Brian, Stephanie, Jace, Jerry	4/6/12	4/6/12	1
16	Advisor Meeting #10	Brian, Stephanie, Jace, Jerry	4/13/12	4/13/12	1
17	Final Breadboard and Testing	Brian, Stephanie, Jace, Jerry	4/13/12	5/3/12	15
17.1	Sun Simulator	Brian, Stephanie	4/13/12	5/3/12	15
17.2	Solar Collector	Jace, Jerry	4/13/12	5/3/12	15
18	Advisor Meeting #11	Brian, Stephanie, Jace, Jerry	4/20/12	4/20/12	1
19	Advisor Meeting #12	Brian, Stephanie, Jace, Jerry	4/27/12	4/27/12	1
20	Initial Release (End of Spring Semester)	Brian, Stephanie, Jace, Jerry	5/4/12	5/4/12	
21	Order and Receive Mechanical Hardware		4/27/12	8/24/12	86
21.1	Sun Simulator	Brian, Stephanie	4/27/12	8/24/12	86
21.2	Solar Collector	Jace, Jerry	4/27/12	8/24/12	86
22	Project Build		8/24/12	9/20/12	20
22.1	Sun Simulator	Brian, Stephanie	8/24/12	9/20/12	20
22.2	Solar Collector	Jace, Jerry	8/24/12	9/20/12	20
23	Verify and Validate		9/20/12	12/6/12	56
23.1	Sun Simulator	Brian, Stephanie	9/20/12	10/17/12	20
23.2	Solar Collector	Jace, Jerry	10/18/12	12/6/12	36
24	Final Project Release (End of Fall Semester)	Brian, Stephanie, Jace, Jerry	12/7/12	12/7/12	

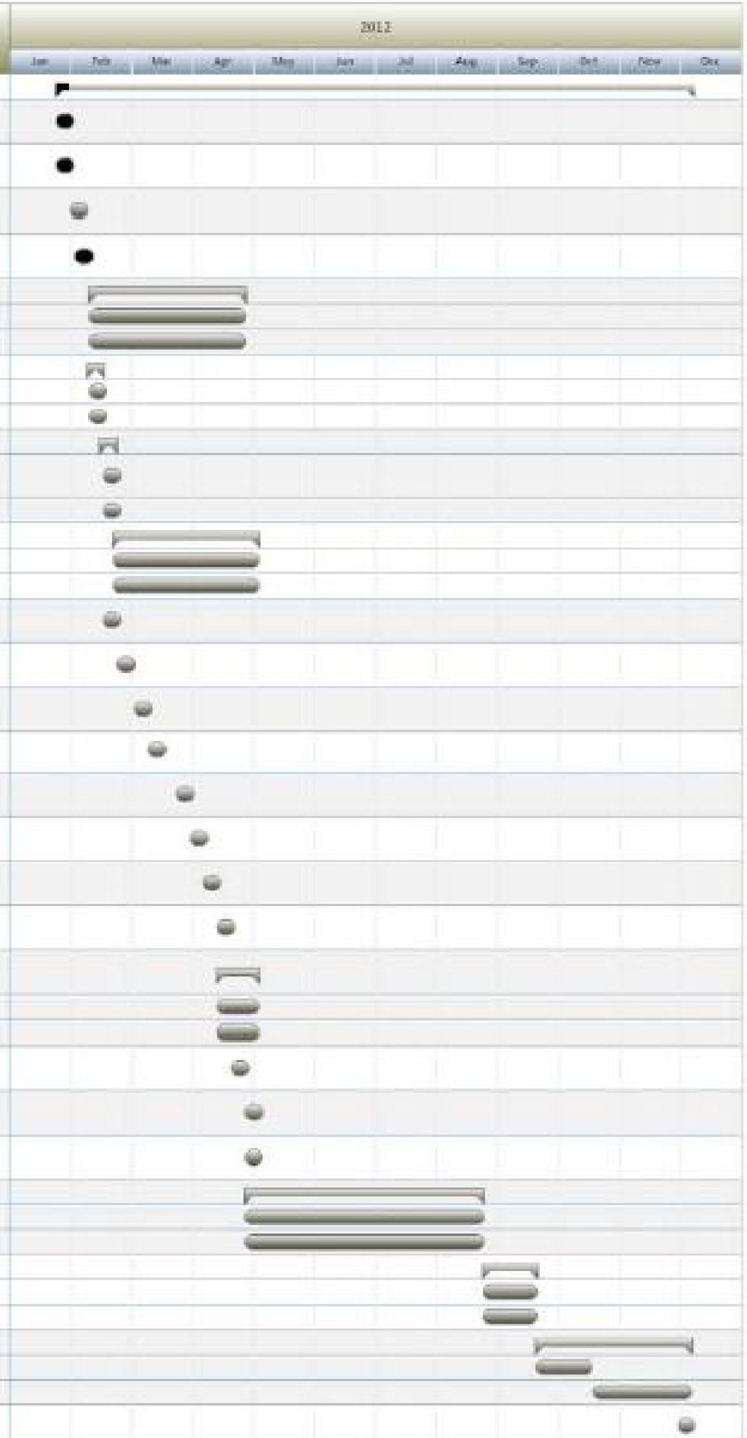


Figure 3: SD1203 Timeline (Design II)

Design III

Name	Begin date	End date
Sun Simulator - Final Design (Brian & Stephanie))	8/27/12	9/5/12
Solar Collector - Final Design (Jace & Jerry)	8/27/12	9/5/12
Sun Simulator - Order Parts (Brian & Stephanie)	9/5/12	9/7/12
Solar Collector - Order Parts (Jace & Jerry)	9/5/12	9/7/12
Apply Mylar to Satellite Dish (Brian)	9/10/12	9/10/12
Breadboard and Test Thermistor Circuit (Brian)	9/10/12	9/14/12
Breadboard and Test Sun Simulator Motor Circuit (Brian)	9/10/12	9/14/12
Breadboard and Test Dimmer Switch Circuit PCB (Stephanie)	9/10/12	9/21/12
Breadboard and Test Solar Collector Motor Circuit PCB (Jace)	9/10/12	9/21/12
Test Inverter and add fuse (Jace)	9/10/12	9/14/12
Breadboard and Test Battery Charger and Protective Circuit PCB (Jerry)	9/10/12	9/21/12
Design Thermistor Circuit PCB in Software (Brian)	9/14/12	9/19/12
Design Sun Simulator Motor Circuit PCB in Software (Brian)	9/14/12	9/19/12
Design Dimmer Switch Circuit PCB in Software (Stephanie)	9/21/12	9/26/12
Design Solar Collector Motor Circuit PCB in Software (Jace)	9/21/12	9/26/12
Design Battery Charger and Protective Circuit PCB in Software (Jerry)	9/21/12	9/26/12
Order Thermistor Circuit PCB (Brian)	9/21/12	9/21/12
Order Sun Simulator Motor Circuit PCB...	9/21/12	9/21/12
Order Dimmer Switch Circuit PCB (Stephanie)	9/26/12	9/26/12
Order Solar Collector Motor Circuit PCB (Jace)	9/26/12	9/26/12
Order Battery Charger and Protective Circuit PCB (Jerry)	9/26/12	9/26/12
Assemble Water Pump System (Stephanie)	9/21/12	9/28/12
Assemble Sun Simulator (Brian and Stephanie)	9/21/12	10/5/12
Assemble Solar Collector (Jace and Jerry)	9/21/12	10/5/12
Populate and Assemble Thermistor PCB (Brian)	9/27/12	9/28/12
Populate and Assemble Sun Simulator Motor PCB (Brian)	9/27/12	9/28/12
Populate and Assemble Dimmer Switch Circuit PCB (Stephanie)	10/3/12	10/8/12
Populate and Assemble Solar Collector Motor Circuit PCB (Jace)	10/3/12	10/8/12
Populate and Assemble Battery Charger and Protective Circuit PCB (Jerry)	10/3/12	10/8/12
Test and Trouble Shoot Sun Simulator (Brian and Stephanie)	10/5/12	10/19/12
Test and Trouble Shoot Solar Collector (Jace and Jerry)	10/5/12	10/19/12
Redesign and Test Sun Simulator if neccesary (Brian and Stephanie)	10/19/12	11/2/12
Redesign and Test Solar Collector if neccesary (Jace and Jerry)	10/19/12	11/2/12
Sun Simulator Final Testing (Brian and Stephanie)	11/2/12	11/14/12
Solar Collector Final Testing (Jace and Jerry)	11/2/12	11/14/12
Demo Day	11/15/12	11/15/12

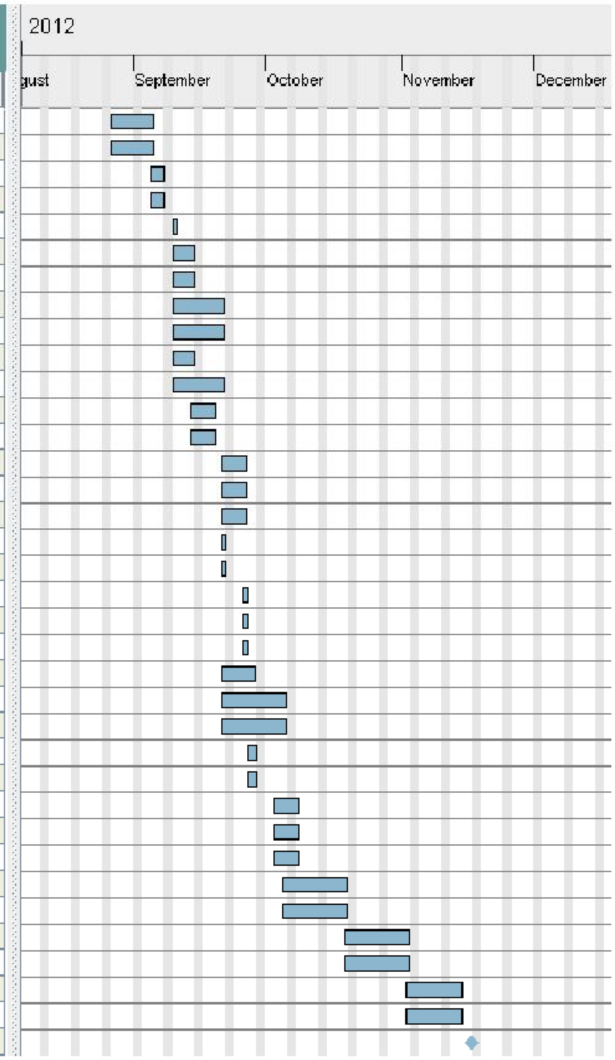


Figure 4: SD1203 Timeline (Design III)

Pictures and Operation

Solar Tracker

The Solar tracker provides a platform which will have the following basics functions:

- Track the sun at a 90 degree angle at all-time throughout the day
- Shuts off automatically when light intensity is below a set threshold, and turn back on when light intensity rises back up.

We can attach anything system on this platform that requires facing the sun at a right angle. For our senior design project, we intended to use this platform for either a solar panel or a satellite dish system.

The solar panel is used to collector sunlight and charge a 12V battery. The satellite dish is used to convert sunlight into heat and release the heat through a heat exchanger for experimental purposes.

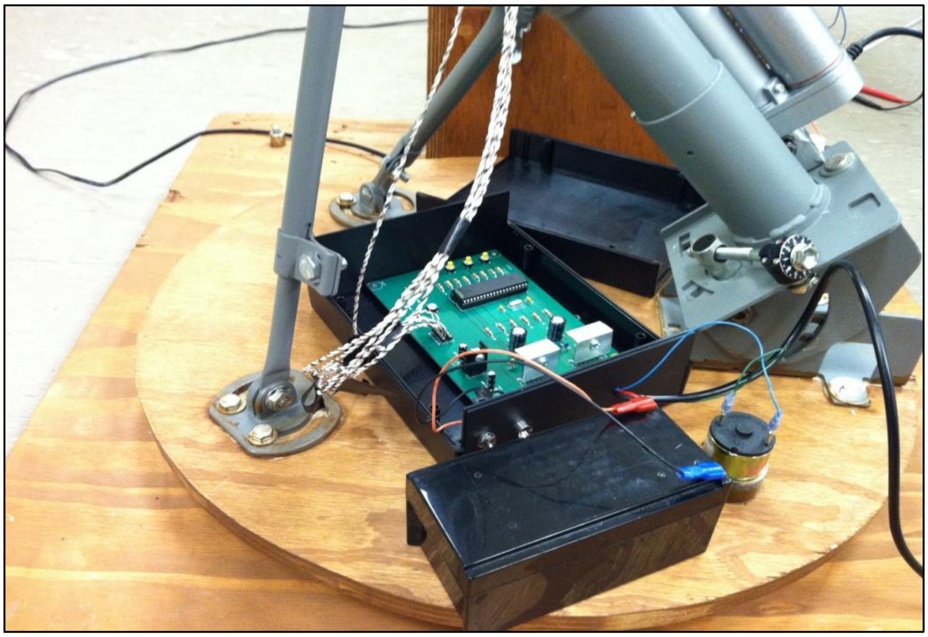


Figure 5: Solar Tracker Base

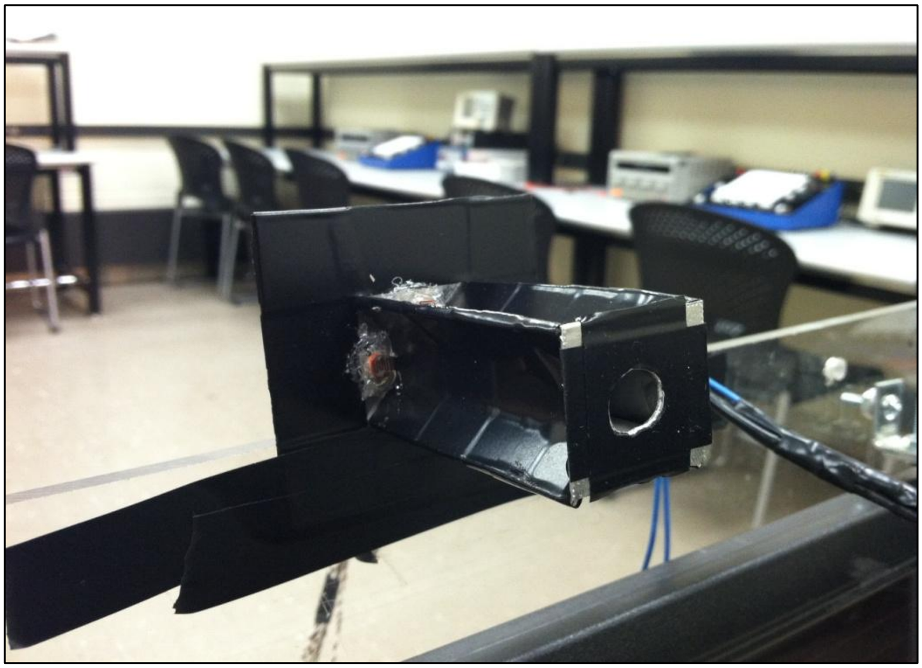


Figure 6: Solar Tracker Light Sensors

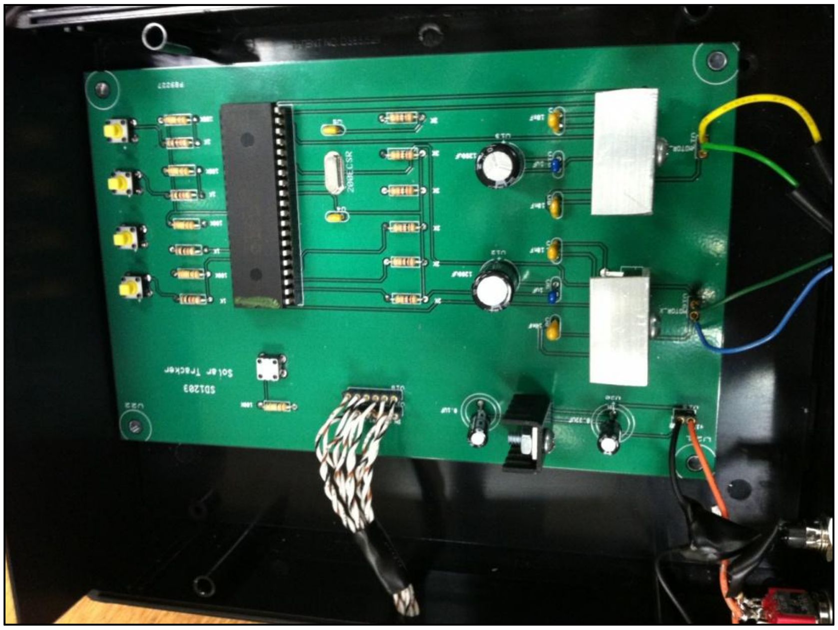


Figure 7: Solar Collector PCB

Solar Panel System

The solar panel system is intended to be self-sustaining; meaning the energy collected will charge a 12-volt battery which will run the solar tracker system. This will be accomplished by having one battery (charged) run the system, and one battery being charged in the process. The batteries will be interchanged when necessary. The charged battery will also run a 300-watt inverter so common household appliances can be used.

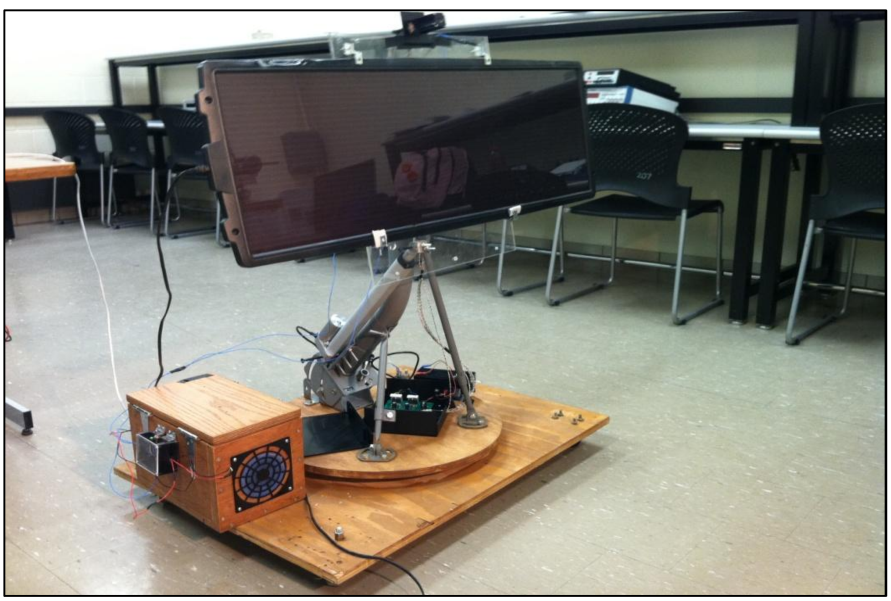


Figure 8: Solar Tracker with Solar Panel

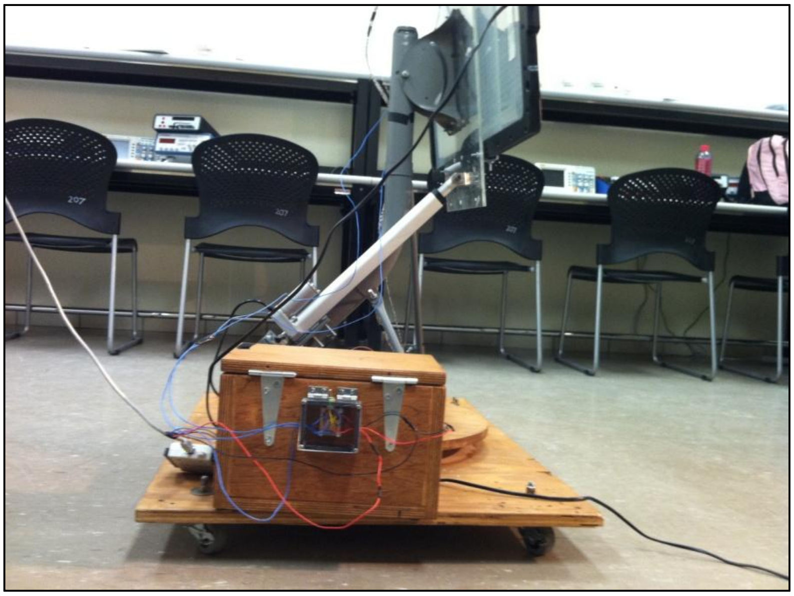


Figure 9: Solar Collector with Solar Panel Installed

Satellite Dish System

The satellite dish system contains a satellite dish covered with a reflective Mylar material installed on the Solar Tracker. As the Solar tracker tracks the sun the reflective Mylar material will focus all the sunlight to the dish's focal point. A hollow copper coil tube, which contains continuously running fluid, will be placed at the focal point. As the sunrays are reflected and focused at the copper coil tube, essentially at one point, the copper coil will begin to heat up which will consequently heat the fluid inside the coil. Thermal insulated tubing will connect the copper coil to the heat exchanger and a tank reservoir, so that the heat from the liquid can be measured. The system is powered using a charged 12-volt battery.

Solar Tracker Operation

It is very simple to operator this system once the solar panel is installed on the platform. The only operation button is the power switch on the PCB enclosure; the power indicator will indicate whether the system is turned on or not. Once it is turned on, it will start its full automatic operations and track the sun at all time.

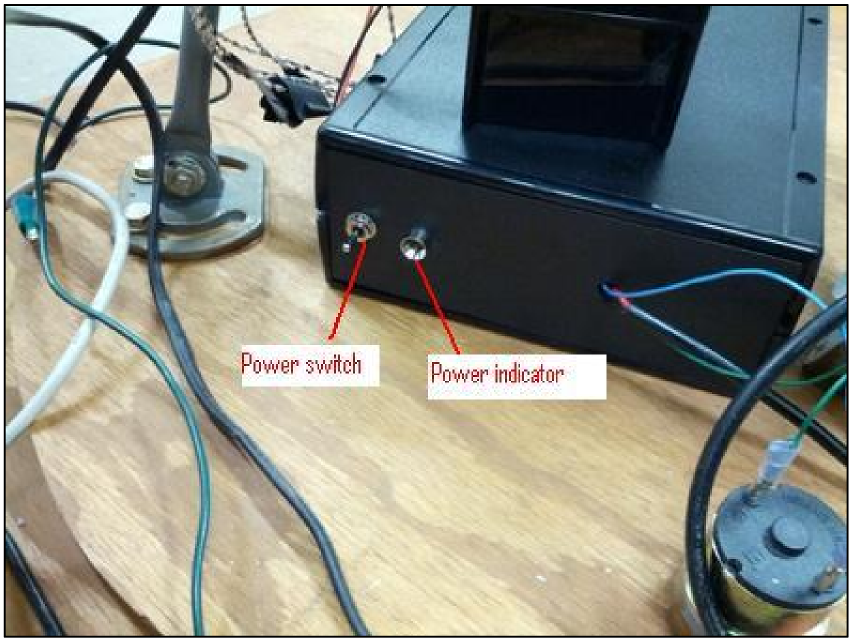


Figure 10: Power switch and power indicator

Both the solar panel and satellite dish can be mounted on the platform provided on the solar tracking system. The solar panel is connected to a voltage regulating recharging circuit to charge a 12V battery, and the satellite dish is connected to a heat exchanger. Both will be shown in the next sections.


Figure 11: Solar tracker platform

Heat Exchanger

The heat exchanger is used to contain and measure the heat generated with the satellite system attached. As the water is heated the coil inside the box begins to heat which in turn heats the inside of the box through the process of convective heat transfer. Using a NI USB 6008 DAQ, the temperature data will be plotted in real time using LabVIEW. The DAQ provides +5V (pin 31) to power the heat exchanger circuit. This only occurs when the DAQ is plugged into the computer USB port.

Before using the heat exchanger, verify all hardware and software are properly connected. Before operation verify the DAQ is properly connected to the heat exchanger circuit by verifying the ground wire (black wire) is connected to port 1, the +5V wire (red wire) is connected to port 31, and the analog input wire (blue wire) is connected to port 2. Please refer to figure 52 in the technician troubleshooting section for the DAQ pinout.

The +5V to power the heat exchanger is provided by the DAQ. The power to the heat exchanger circuit can be turned “on” and “off” using the toggle switch. When the heat exchanger circuit is “on” the green LED light will be illuminated. Figure 12 displays the heat exchanger enclosed circuit, toggle switch, and green LED.

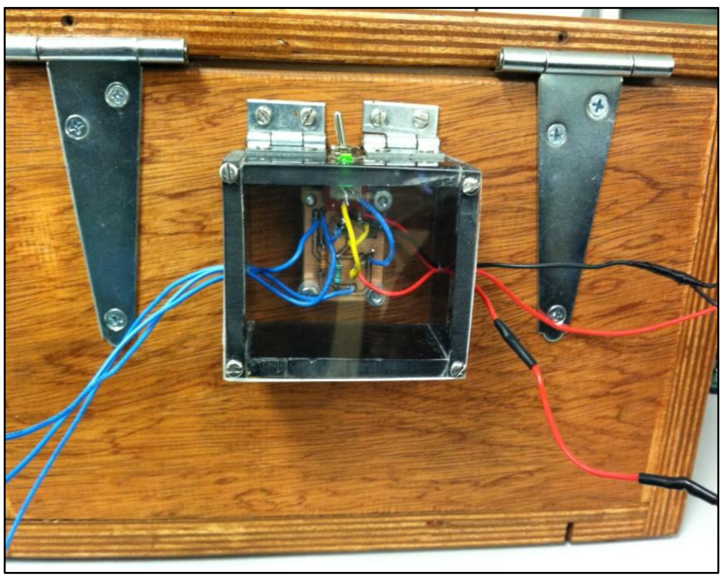
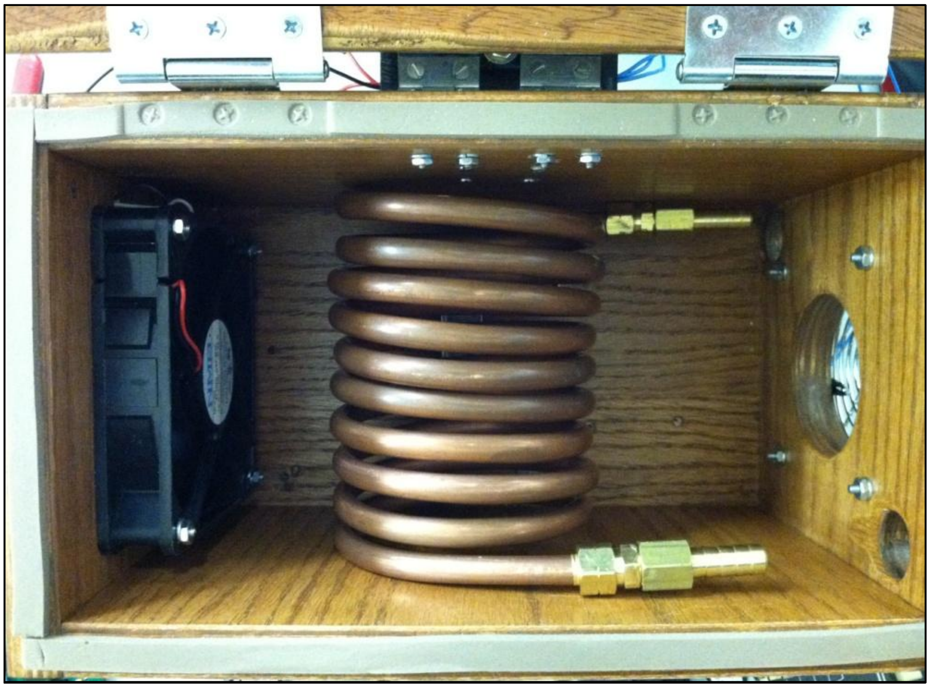


Figure 12: Heat Exchanger Back

Fixture 13: Heat Exchanger Side



Fixture 14: Heat Exchanger Top (open)

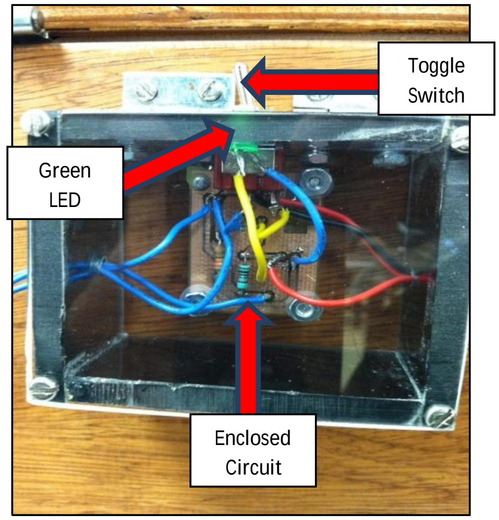
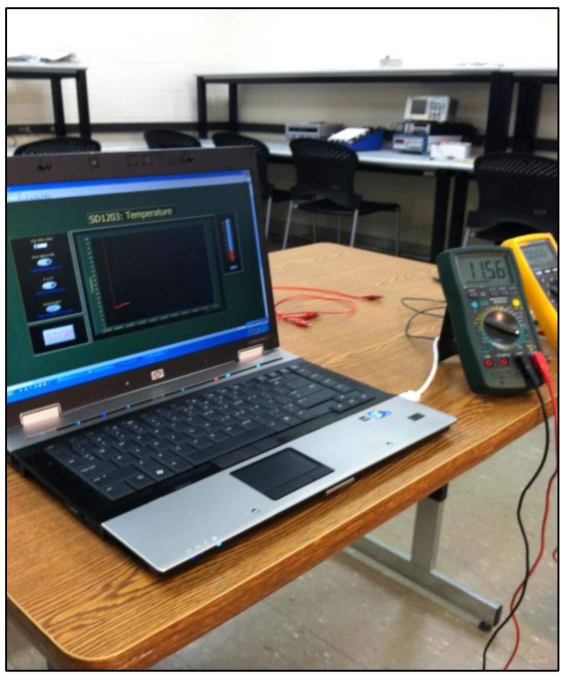


Figure 15: Heat Exchanger Circuit, Toggle Switch, and Green LED



Fixture 16: Heat Exchanger Program on Computer

Battery Charger

We used a buck-boost IC to build a recharger system. This system can serve the Solar Tracker because the Solar Tracker needs a DC voltage supply. The Buck-boost IC can supply a stable voltage to charge the battery. The only problem is the design uses the MC33063A IC, which will only work when the supply voltage is higher than 3V. This supply voltage is provided from the solar panel. As we know, we cannot make sure the voltage supply is greater than 3V because the intensity of illumination outside is uncontrollable. A diode was added between the battery

and the voltage output to prevent the current from flowing back into the system, causing damage to the system. The output voltage was designed to be 14.5V, providing an output current of about 0.225A. The range of the MC33063A IC input voltage is large, ranging from 3V to 40V. This system is easy to modify when different output voltages are needed, because the output value is only dependant on R1 and R2 (refer to equation 1).

$$V_{out} = -1.25x(1 + \frac{R_2}{R_1})$$

Equation 1: Output Voltage of the Charger Circuit

Sun Simulator

The mechanical fixture is composed of $\frac{3}{8}$" steel pipe, a combination of pulleys, stepper motor, and a linear actuator to drive the heat bulb to mimic the suns movement. A 250W infrared heat lamp is used to represent the sun and will be the source of heat. A dimmer switch is used with the heat lamp, which dims the heat lamp to represent cloud cover. The dimmer switch is controlled using a PIC microcontroller and Pulse Width Modulation (PWM). As stated above, temperature data is collected and plotted in LabVIEW. With the dimming of the lamp, a decrease in temperature should be evident in the LabVIEW plotted data. Both motor control and dimmer switch is controlled by a PIC microcontroller, and the programming was done in C. Figure 17 displays the Sun Simulator and all of it components.

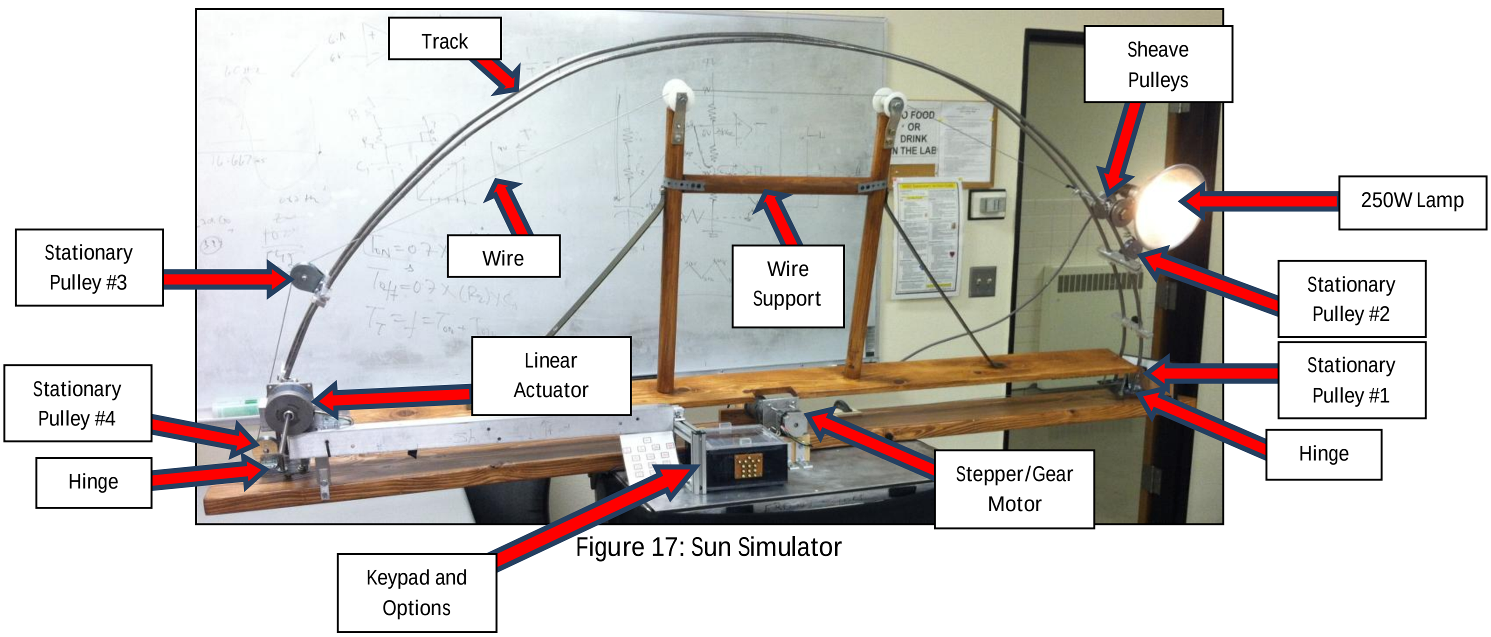


Figure 17: Sun Simulator

Power

First, locate the power source on the cart behind the black curtain. The first time using the simulator, make sure the Sun Simulator is NOT connected to the power source. Once the simulator is removed, turn on the power source by pressing the green power button. When the power is "on" verify the

voltage level is set to +5V. When the power supply voltage is not set to +5V, turn the voltage knob to set the power supply to +5V. Once the power supply is set to +5V turn the power supply off and re-connect the Sun Simulator. Once connected turn the power supply back "on" and now the Sun Simulator is ready to simulate. NOTE: The Sun Simulator PCB can only run with +5V, voltages higher than +5V may damage the PCB.

Keypad and Keypad Options for Motor Operation

The Sun Simulator motors are operated from the keypad located on the PCB enclosure. Keypad options are located directly to the left of the keypad, please see figure 18.

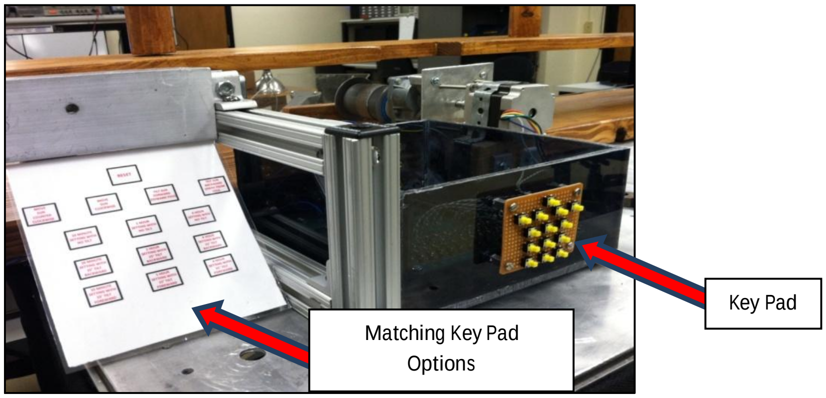


Figure 18: Key Pad and Key Pad Options

NOTE: The Sun Simulator lamp starting position must always start on the right (when facing the simulator front). Verify the black electrical tape is directly over "stationary pulley #2." This movement represents the sun rising in the east and setting in the west. If this is not verified on the initial run, the sheave pulley will go beyond "stationary pulley #3" and could potentially damage the simulator. Please refer to figure 17 for locations of stationary pulleys.

In a similar fashion, if you plan to use a preset option involving a tilt you must verify the track is completely vertical in the starting position (90° on the protector display, see figure 19). Failure to do this could cause damage to the track and/or linear actuator.

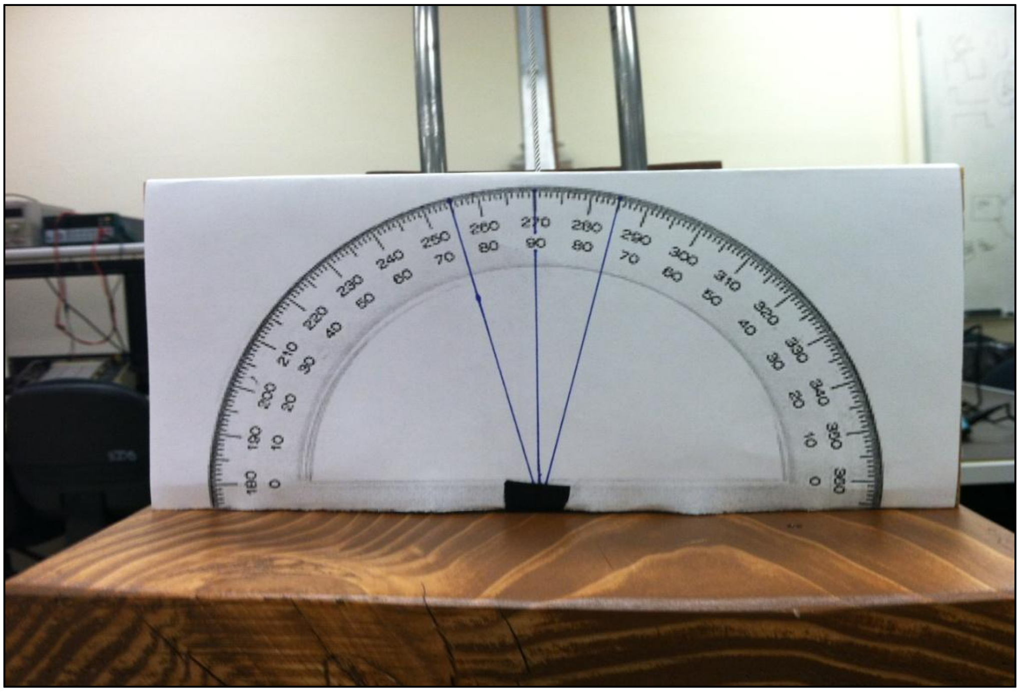


Figure 19: Sun Simulator track in starting position, no tilt

Figure 20 is the keypad options, and a description of each button is provided below.

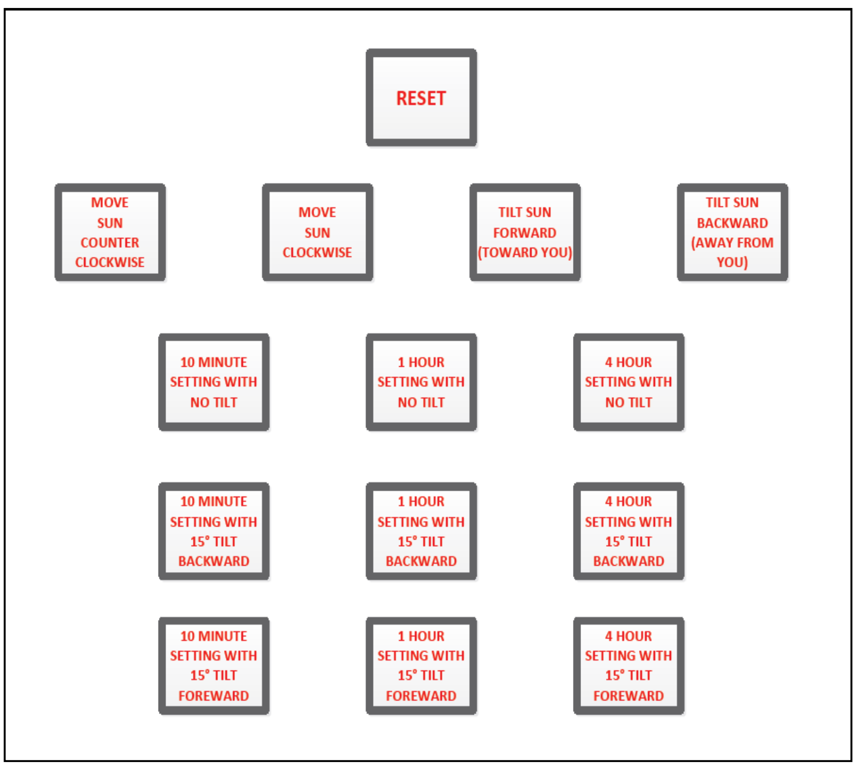


Figure 20: Sun Simulator Key Pad Options

RESET: Pressing the “RESET” button resets the microcontroller. If the reset button is pressed during a run, the lamp will have to be manually moved (using the manual buttons described next) back to the beginning before another run is started. Refer to the note above.

MOVE SUN SIMULATOR CLOCKWISE: If facing the Sun Simulator front, pressing this option moves the lamp to the right (sun moving west to east). This button is often used to manually move the lamp to the beginning position if reset is pressed or the power was turned off.

MOVE SUN SIMULATOR COUNTERCLOCKWISE: If facing the Sun Simulator front, pressing this option moves the lamp to the left (sun moving east to west).

TILT SUN FORWARD (TOWARD YOU): This option tilts the track towards you, if you are facing the front of the simulator.

TILT SUN BACKWARD (AWAY FROM YOU): This option tilts the track away from you, if you are facing the front of the simulator.

10 MINUTE SETTING WITH NO TILT: This preset option runs the lamp (sun) right to left (east to west) in a 10 minute interval. This option does not tilt the track, so this option can be run if the track is already tilted. NOTE: This option requires the lamp to be in the starting position.

1 HOUR SETTING WITH NO TILT: This preset option runs the lamp (sun) right to left (east to west) in a 1 hour interval. This option does not tilt the track, so this option can be run if the track is already tilted. NOTE: This option requires the lamp to be in the starting position.

4 HOUR SETTING WITH NO TILT: This preset option runs the lamp (sun) right to left (east to west) in a 4 hour interval. This option does not tilt the track, so this option can be run if the track is already tilted. NOTE: This option requires the lamp to be in the starting position.

10 MINUTE SETTING WITH 15° TILT BACKWWARD: This preset option first tilts the track (sun) 15° backwards (away from you) and then runs the lamp (sun) right to left (east to west) in a 10 minute interval. NOTE: This option does tilt the track, so this option must have the track in the starting position, please see figure 19.

1 HOUR SETTING WITH 15° TILT BACKWWARD: This preset option first tilts the track (sun) 15° backwards (away from you) and then runs the lamp (sun) right to left (east to west) in a 1 hour interval. NOTE: This option does tilt the track, so this option must have the track in the starting position. NOTE: This option requires the lamp to be in the starting position, please see figure 19.

4 HOUR SETTING WITH 15° TILT BACKWWARD: This preset option first tilts the track (sun) 15° backwards (away from you) and then runs the lamp (sun) right to left (east to west) in a 4 hour interval. NOTE: This option does tilt the track, so this option must have the track in the starting

position. NOTE: This option requires the lamp to be in the starting position, please see figure 19.

10 MINUTE SETTING WITH 15° TILT FOREWARD: This preset option first tilts the track (sun) 15° forwards (towards you) and then runs the lamp (sun) right to left (east to west) in a 10 minute interval. NOTE: This option does tilt the track, so this option must have the track in the starting position. NOTE: This option requires the lamp to be in the starting position, please see figure 19.

1 HOUR SETTING WITH 15° TILT FOREWARD: This preset option first tilts the track (sun) 15° forwards (towards you) and then runs the lamp (sun) right to left (east to west) in a 1 hour interval. NOTE: This option does tilt the track, so this option must have the track in the starting position. NOTE: This option requires the lamp to be in the starting position, please see figure 19.

4 HOUR SETTING WITH 15° TILT FOREWARD: This preset option first tilts the track (sun) 15° forwards (towards you) and then runs the lamp (sun) right to left (east to west) in a 4 hour interval. NOTE: This option does tilt the track, so this option must have the track in the starting position. NOTE: This option requires the lamp to be in the starting position, please see figure 19.

Track Tilting

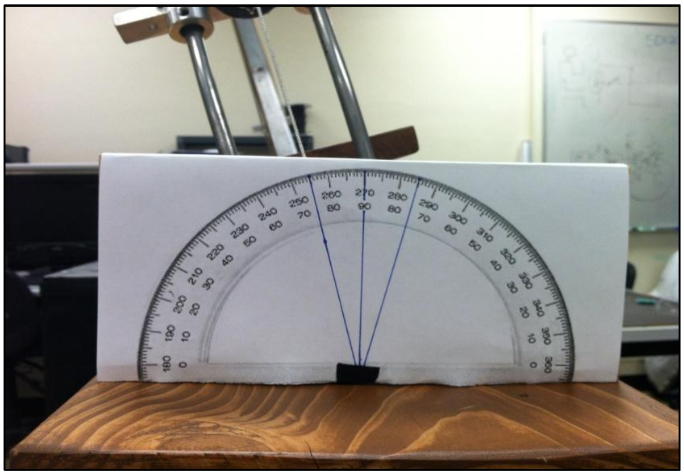


Figure 21: Sun Simulator track with 15° forward

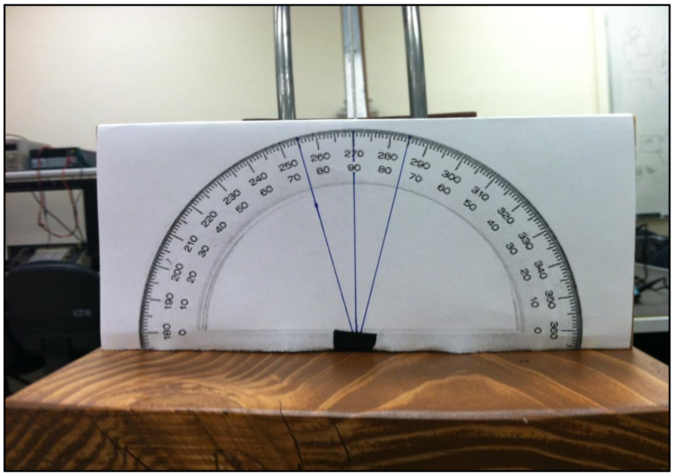


Figure 22: Sun Simulator track in starting position, no tilt

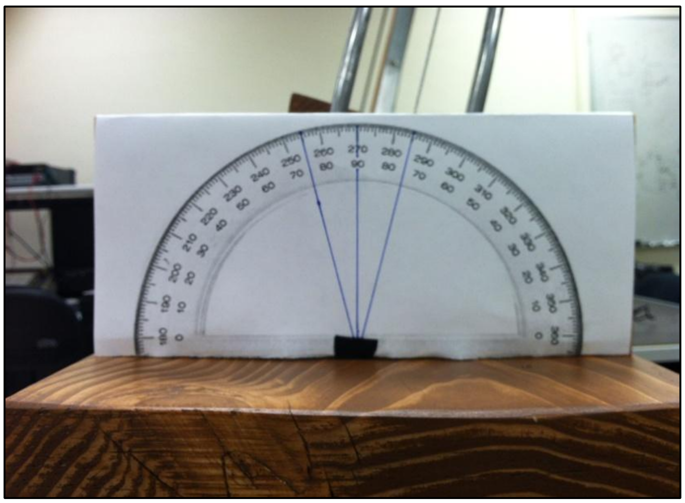


Figure 23: Sun Simulator track with 15° backward

Mechanical Components

Figure 24: Sun Simulator Track, Stationary Pulley, and Sheave Pulley

Figure 25: Sun Simulator Sheave Pulley and Spring Tension

Figure 26: Sun Simulator Wire Support

Motors

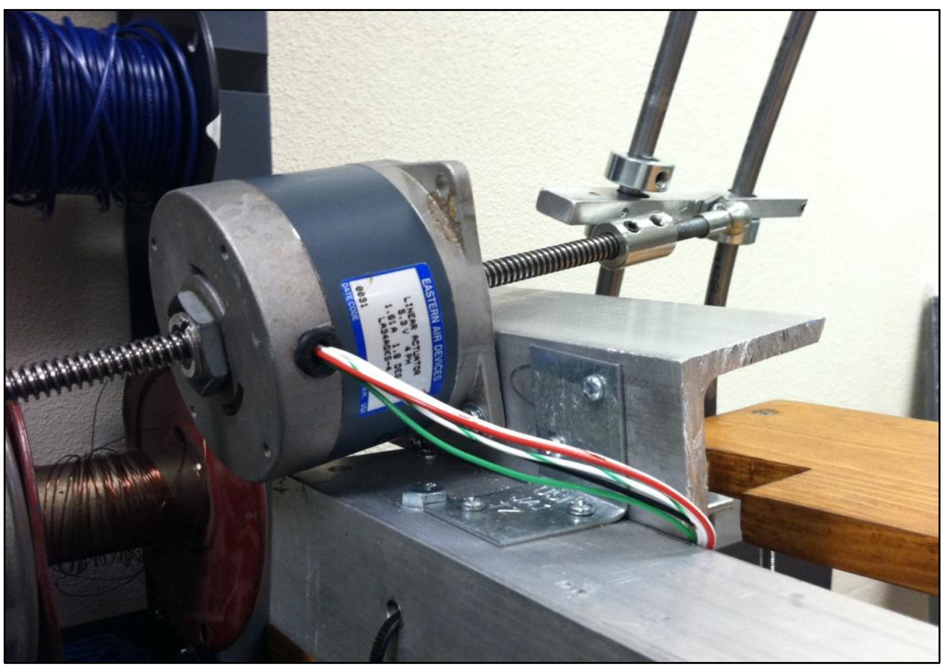


Figure 27: Sun Simulator Linear Actuator (Tilts Track)

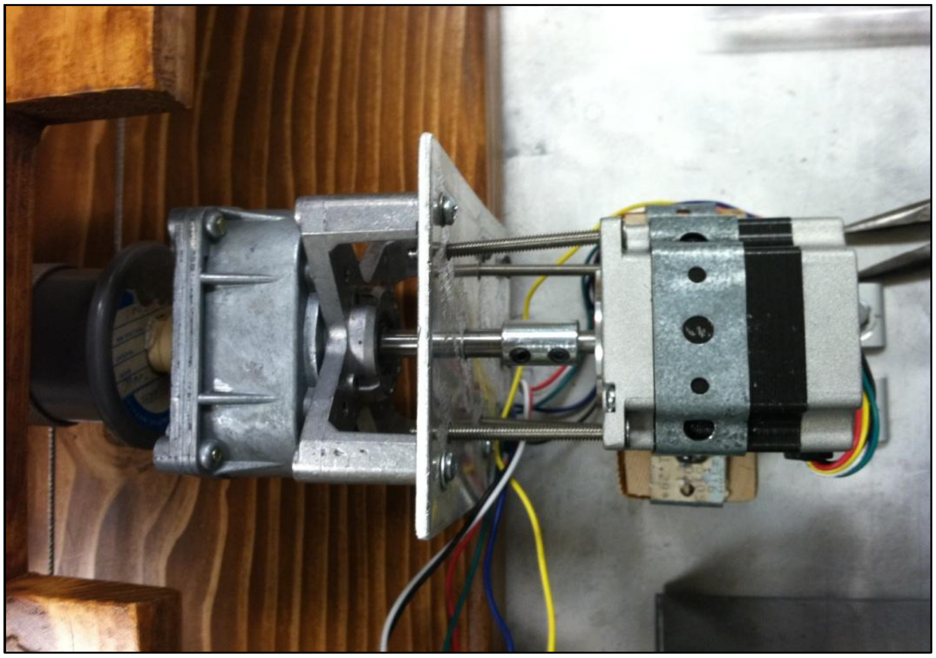


Figure 28: Sun Simulator Stepper motor and Gear Box

Dimmer

The dimmer is used in combination with a 250W infrared heat lamp which represents the sun and is the source of heat for the sun simulator. This switch allows the user to both dim and brighten the light, representing varying amounts of cloud cover. Therefore, this function allows the user to output varying amounts of heat.

The dimmer is controlled using a PIC microcontroller and Pulse Width Modulation (PWM). All programming for this function was done in C. By varying the Pulse Width Modulation output from the PIC microcontroller, the user is able to set the brightness of the heat lamp to the desired level and also simulate gradual increases and decreases in brightness.

Dimmer Operation

The dimmer is operated using a keypad on the PIC microcontroller. The display screen on the PIC microcontroller allows the users to see how much of the total brightness is being outputted by the heat lamp. When turned on, the display will read "PWM = 0.00" and the light will be fully on. "PWM = 0.00" represents that the light is 0% bright. NOTE: "PWM = 1.00" represents that the light is 100% dim; this occurs when the light is completely off.
Each keypad option is provided below:

YELLOW RESET: Pressing the yellow reset button on the PIC microcontroller resets the PIC and allows the user to initialize the Dimmer Switch program. This button must be pressed before the user is able to operate the dimming switch.

RB0: Pressing RB0 incrementally brightens the heat lamp by 5%. Each time this button is pressed, the light will become 5% brighter. If you reach 100% brightness, the output screen will display "PWM = 0.00". This indicates that the light has reached its full potential and pressing R0 will no longer do anything to the light.

RB1: Pressing RB1 incrementally dims the heat lamp by 5%. Each time this button is pressed, the light will become 5% more dim. If you reach 100% dim, the output screen will display "PWM = 1.00". This indicates that the light has fully turned off and pressing RB1 will no longer do anything to the light.

RB2: Pressing RB2 brightens the light incrementally at 1% per 0.1 seconds until the light has become 100% bright. At this point the light will stay fully on until another button is pressed. This button can be pressed at any point in time no matter how bright the light is. It will always start increasing in brightness from the level it was at when RB2 was pressed.

RB3: Pressing RB3 dims the light incrementally at 1% per 0.1 seconds until the light has become 100% dim. (It will be completely off.) At this point the light will stay fully off until another button is pressed. This button can be pressed at any point in time no matter how bright the light is. It will always start decreasing in brightness from the level it was at when RB3 was pressed.

RB4: Pressing RB4 brightens the light incrementally at 1% per second until the light has become 100% bright. At this point the light will stay fully on until another button is pressed. This button can be pressed at any point in time no matter how bright the light is. It will always start increasing in brightness from the level it was at when RB4 was pressed.

R5: Pressing RB5 dims the light incrementally at 1% per second until the light has become 100% dim. (It will be completely off.) At this point the light will stay fully off until another button is pressed. This button can be pressed at any point in time no matter how bright the light is. It will always start decreasing in brightness from the level it was at when RB5 was pressed.

Schematics

Solar Tracker

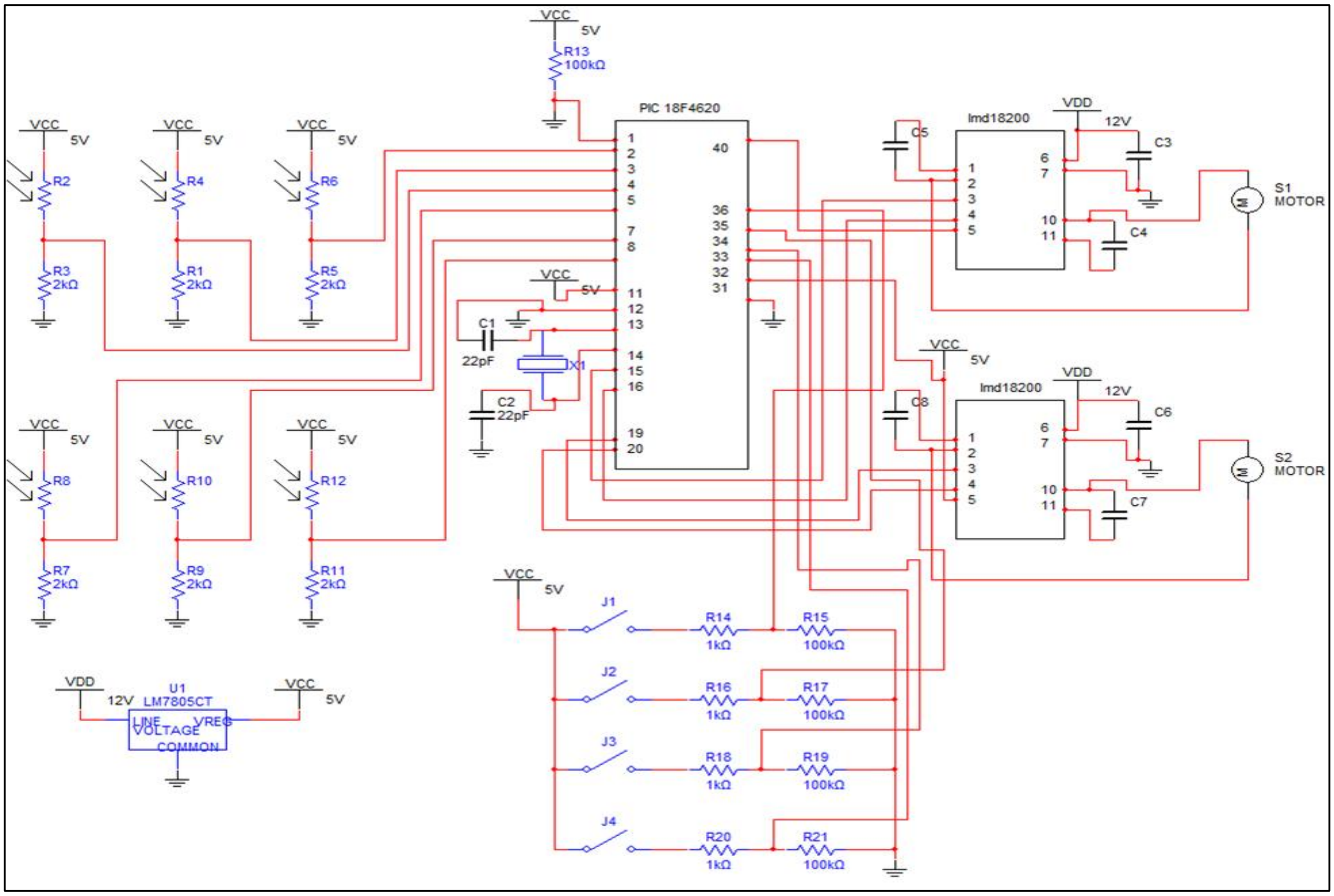


Figure 29: Solar Tracker Schematic

Heat Exchanger

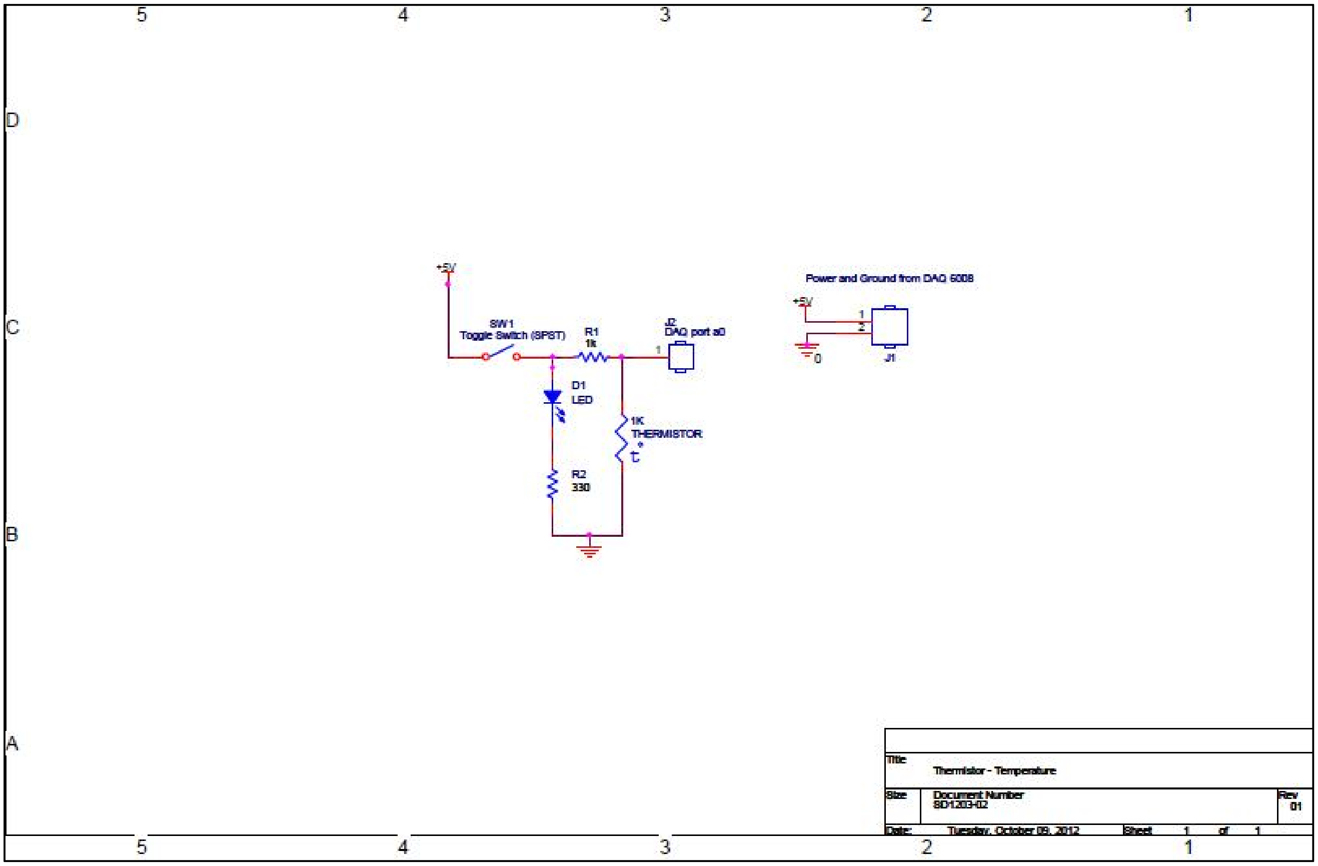


Figure 30: Heat Exchanger Circuit

Battery Charger

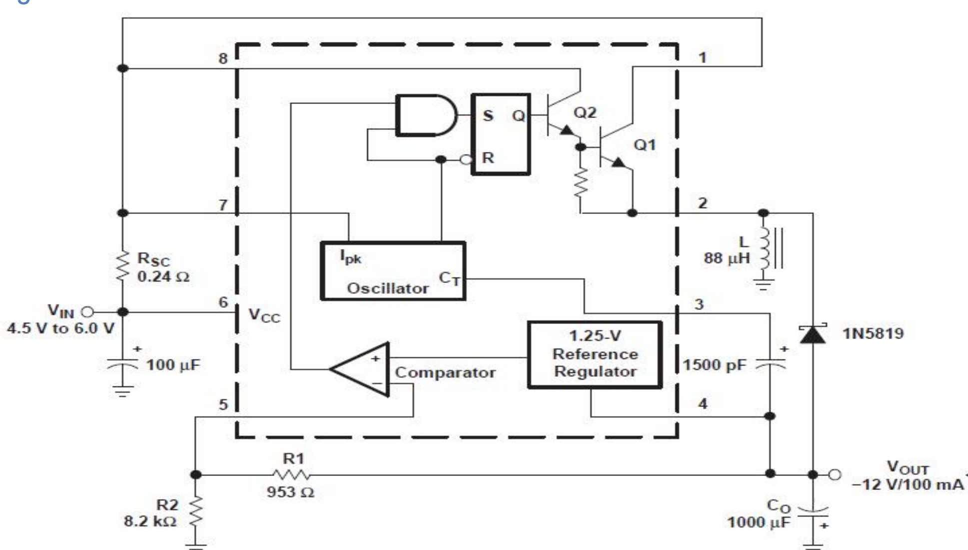


Figure 31: Charger Circuit Schematic

Sun Simulator

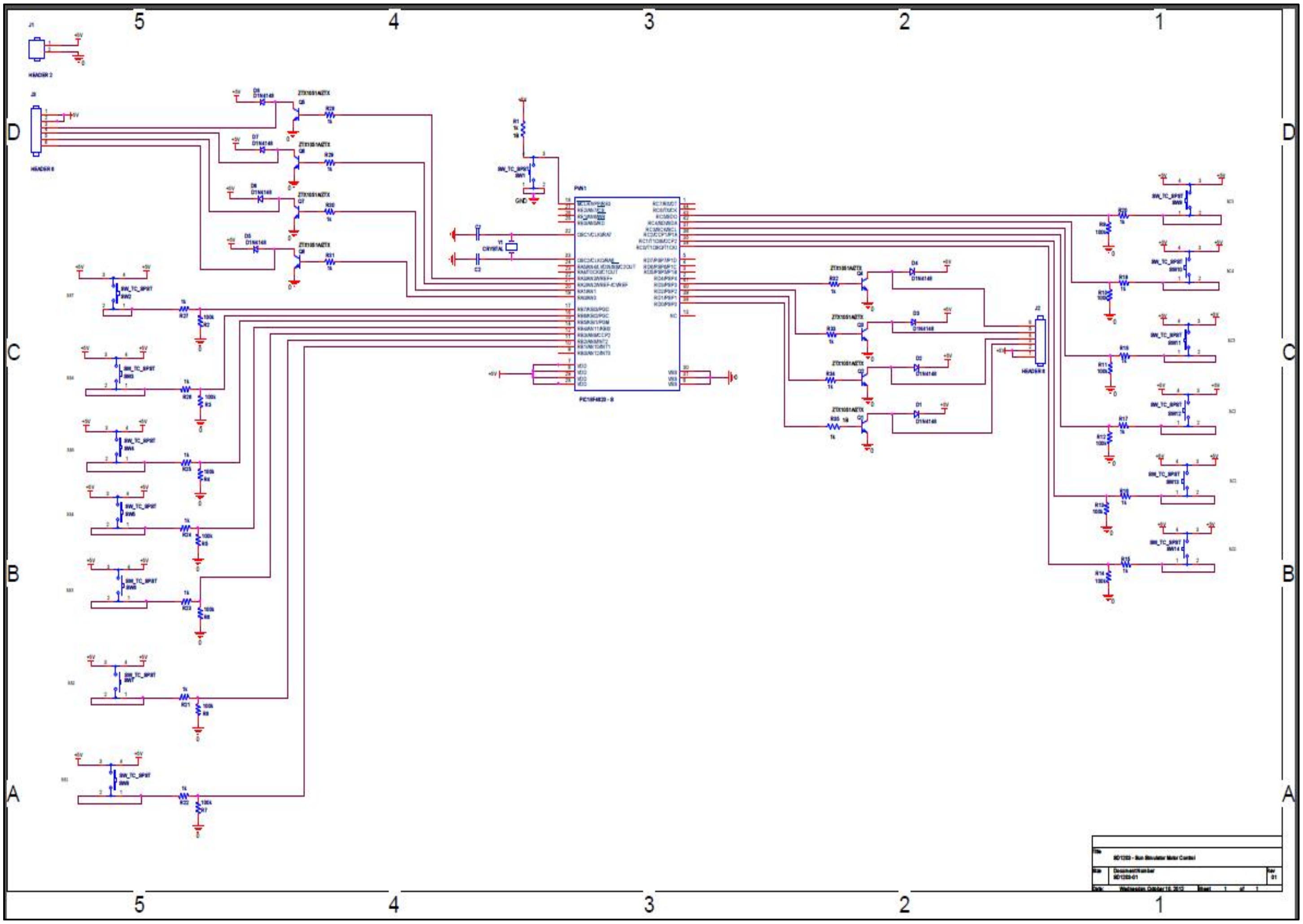


Figure 32: Sun Simulator Motor Control Circuit

Dimmer

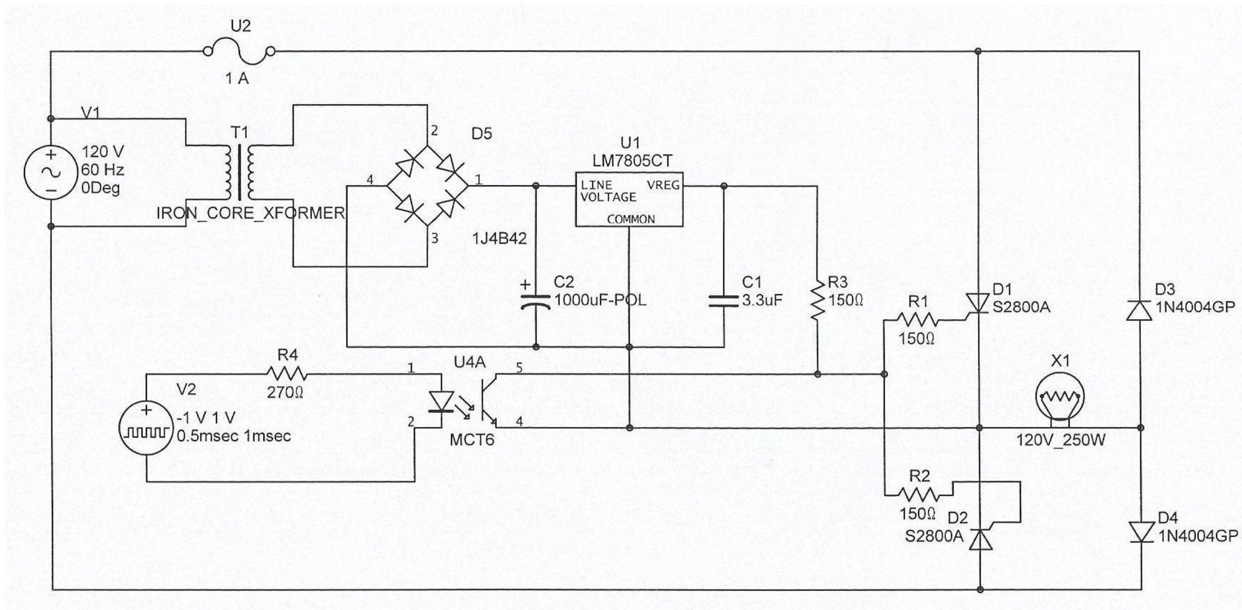


Figure 33: Dimmer Circuit

PCB

Solar Tracker

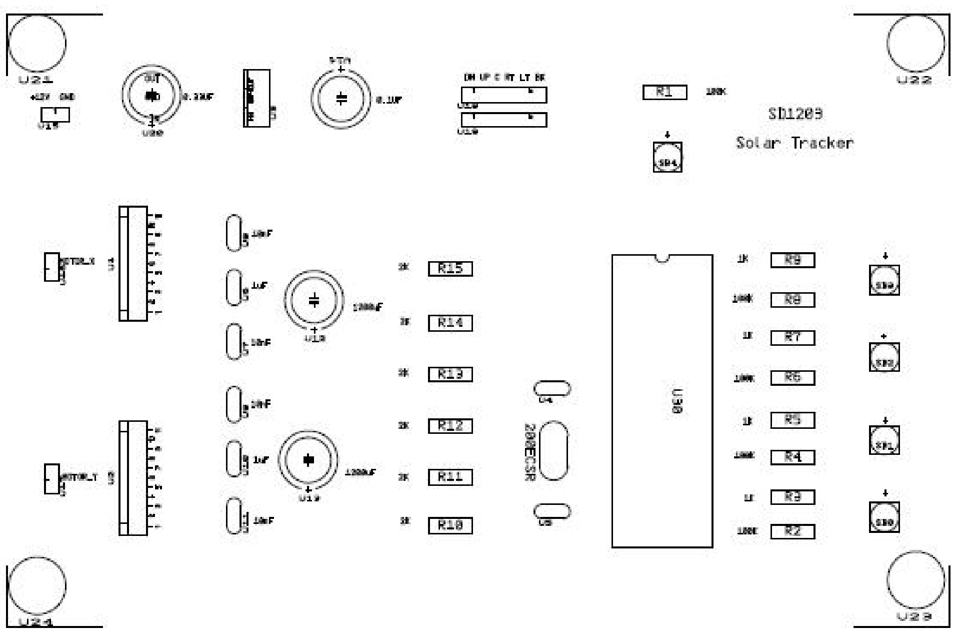


Figure 34: Solar Tracker Silk Screen Top

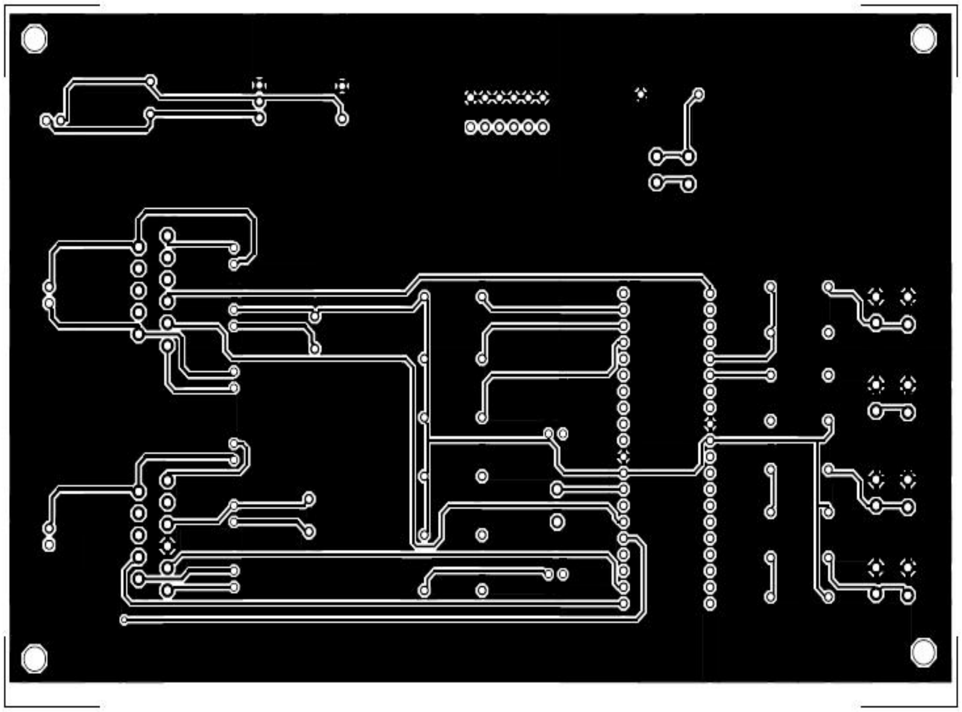


Figure 35: Solar Tracker Copper Top

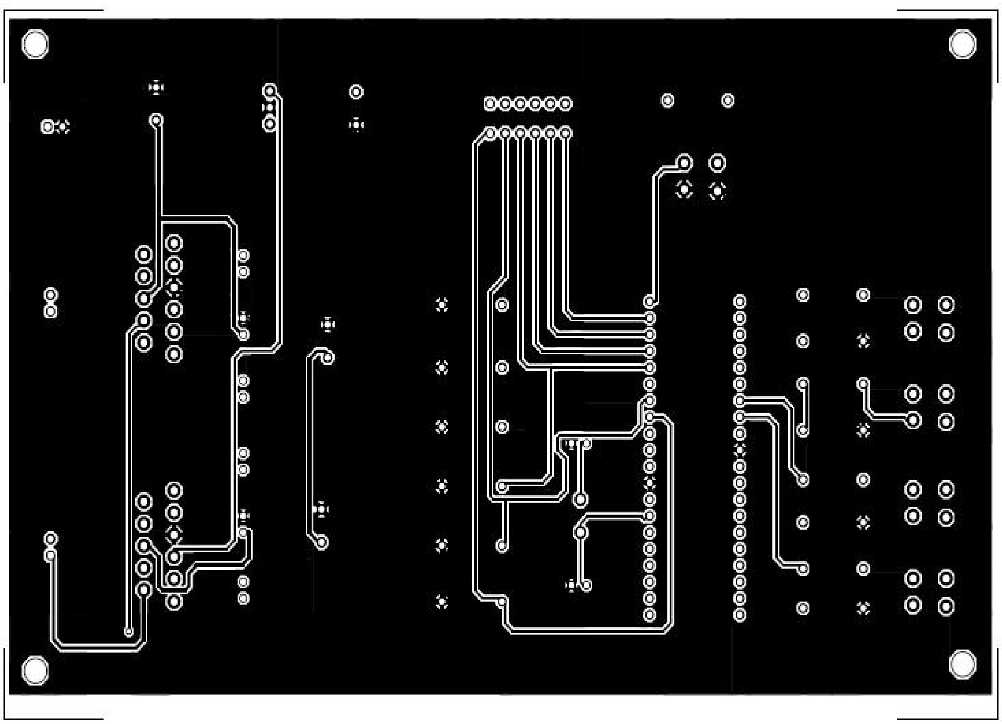


Figure 36: Solar Tracker Silk Copper Bottom

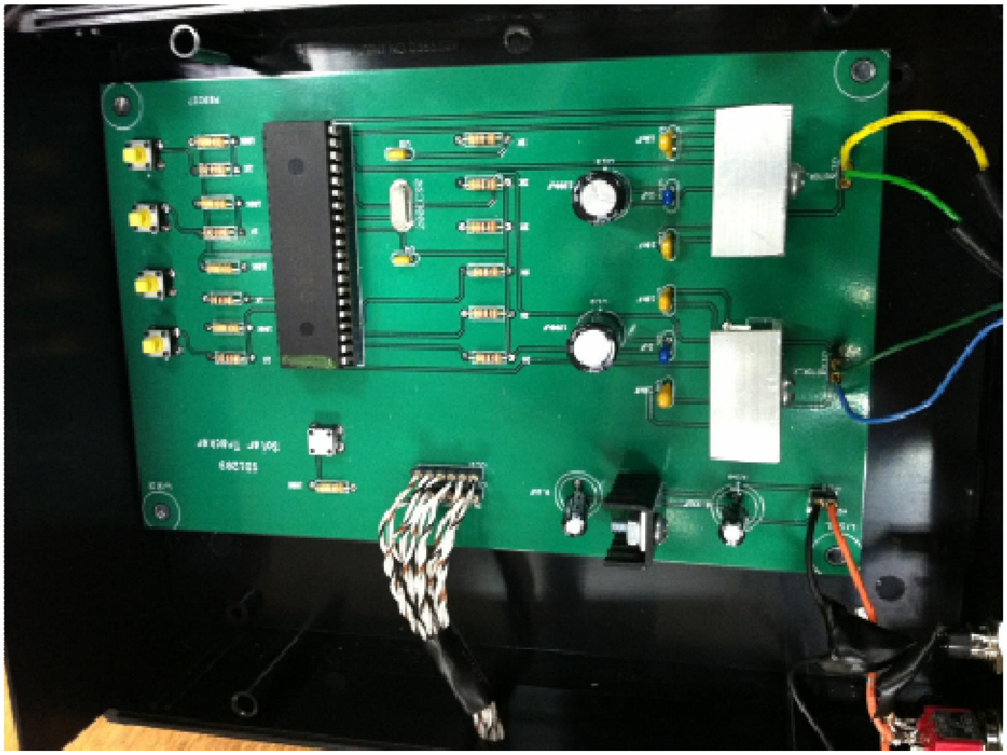


Figure 37: Solar Tracker PCB

Heat Exchanger

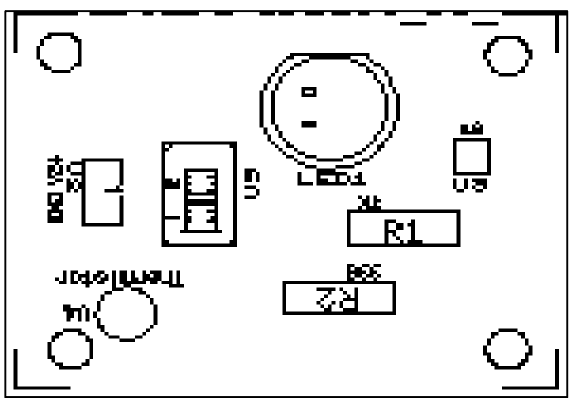


Figure 38: Heat Exchanger Silk Screen Top

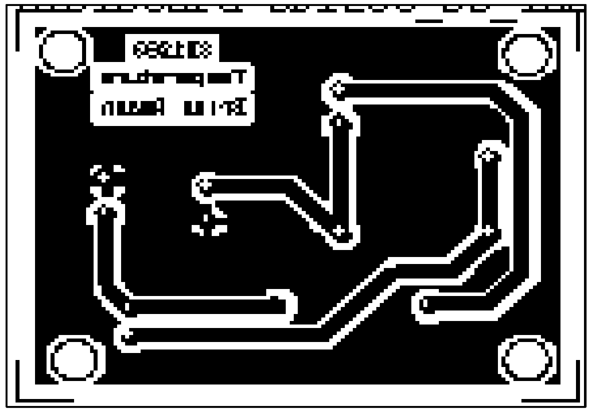


Figure 39: Heat Exchanger Copper Top

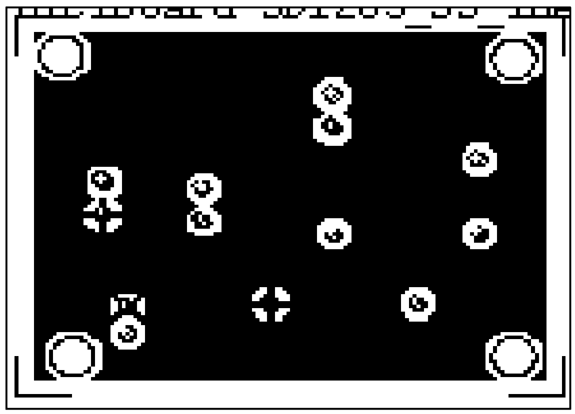


Figure 40: Heat Exchanger Silk Copper Bottom

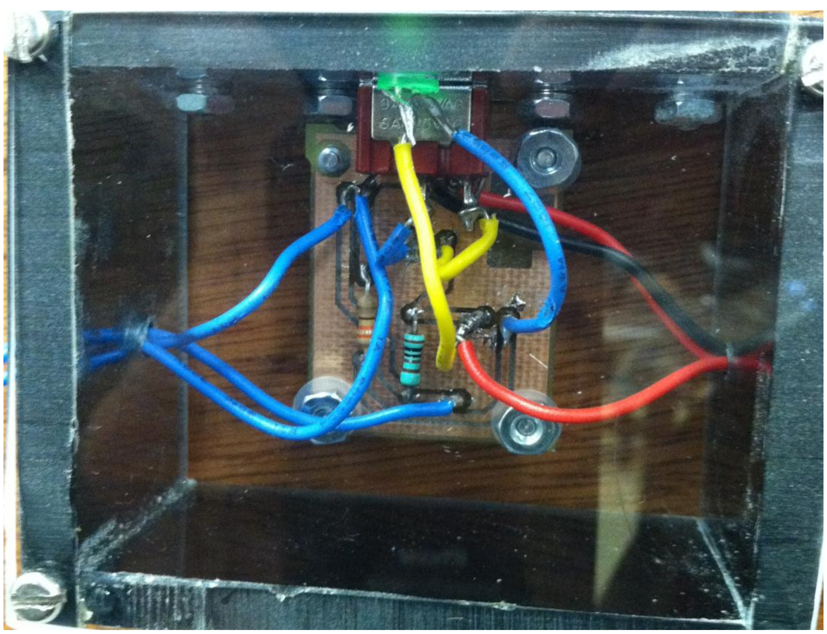


Figure 41: Heat Exchanger PCB

Battery Charger

Insert PCB photos here

Sun Simulator

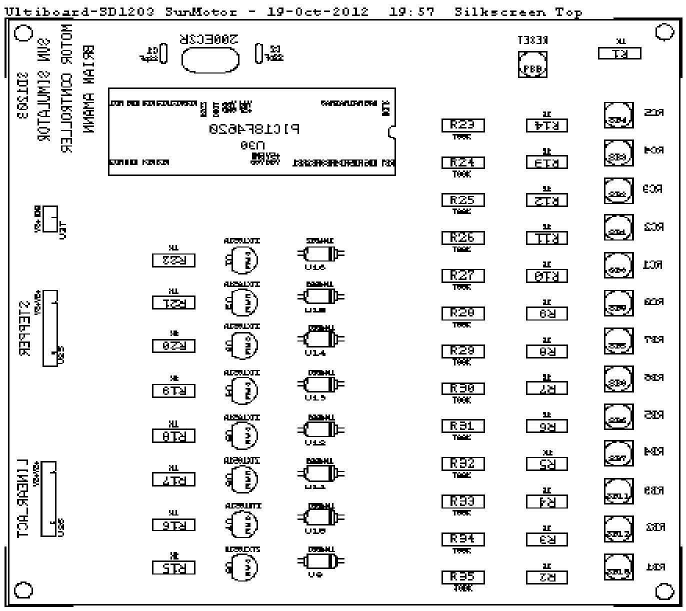


Figure 42: Sun Simulator Silk Screen Top

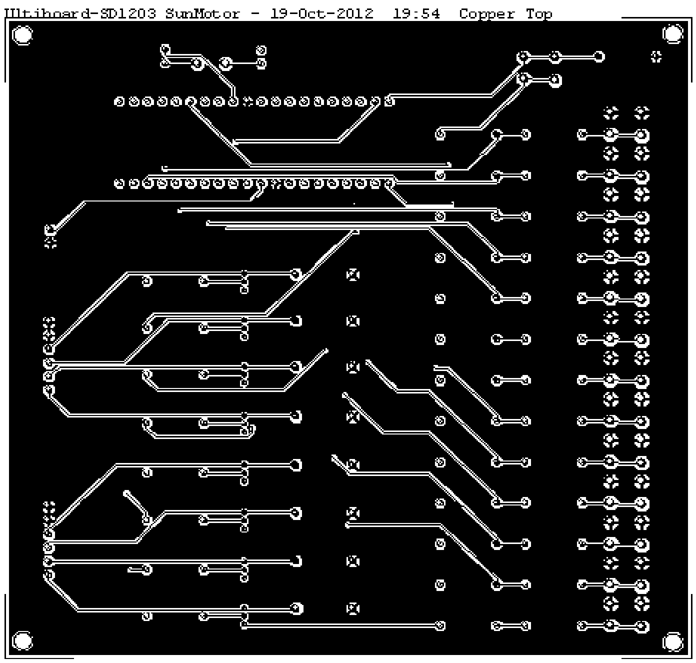


Figure 43: Sun Simulator Copper Top

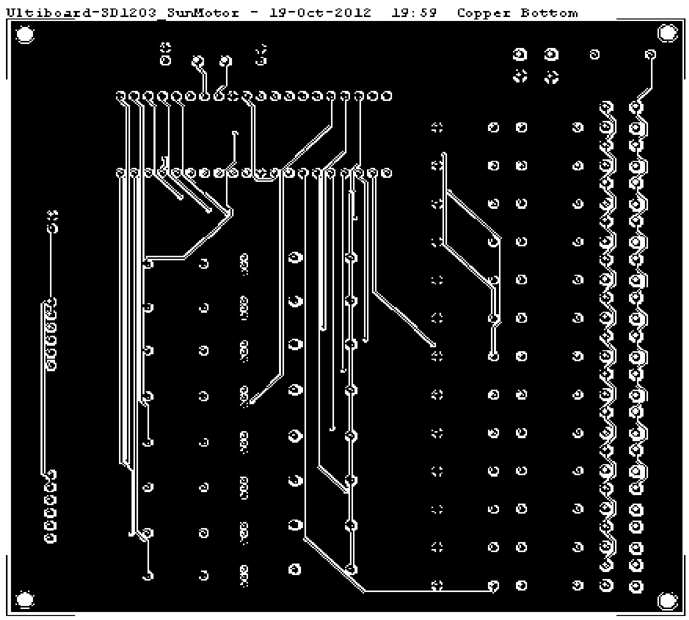


Figure 44: Sun Simulator Copper Bottom

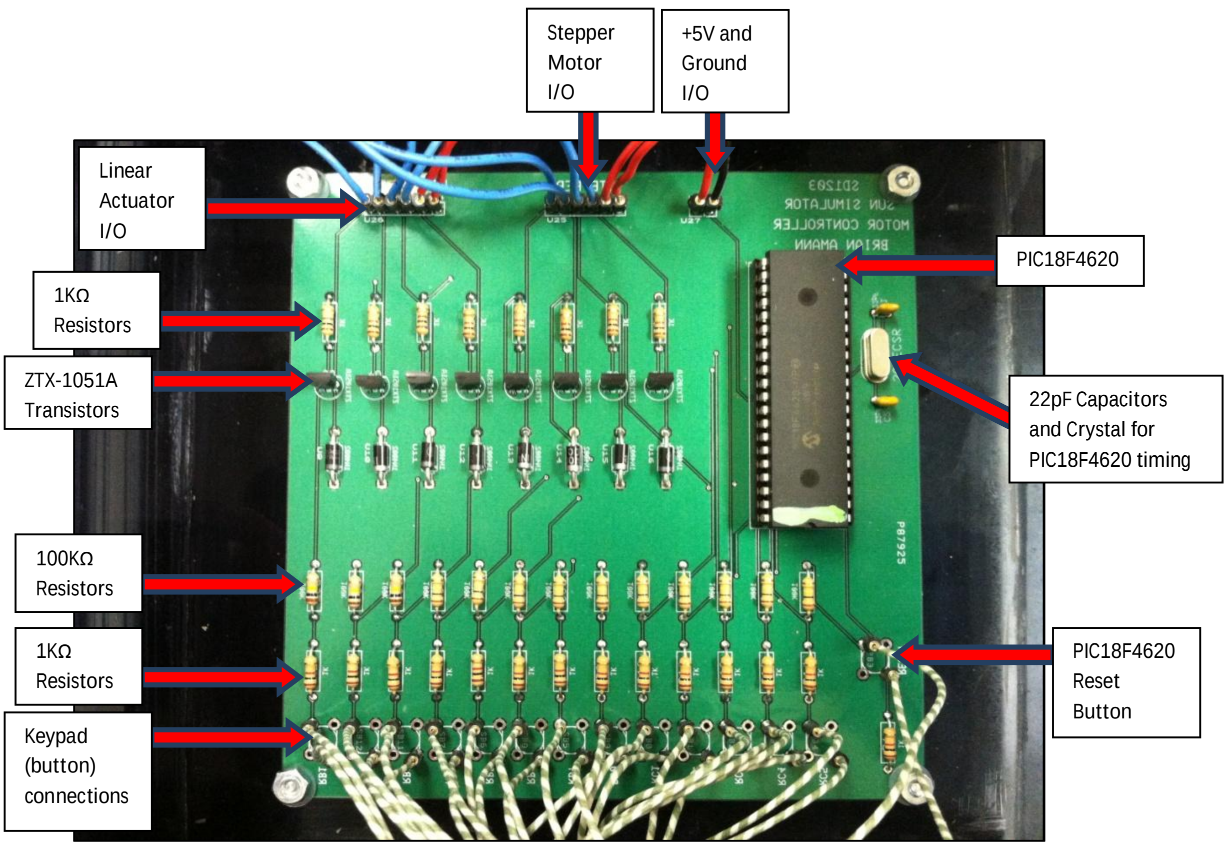


Figure 45: Sun Simulator PCB

Dimmer Switch

Insert PCB photos here

Software

Solar Tracker

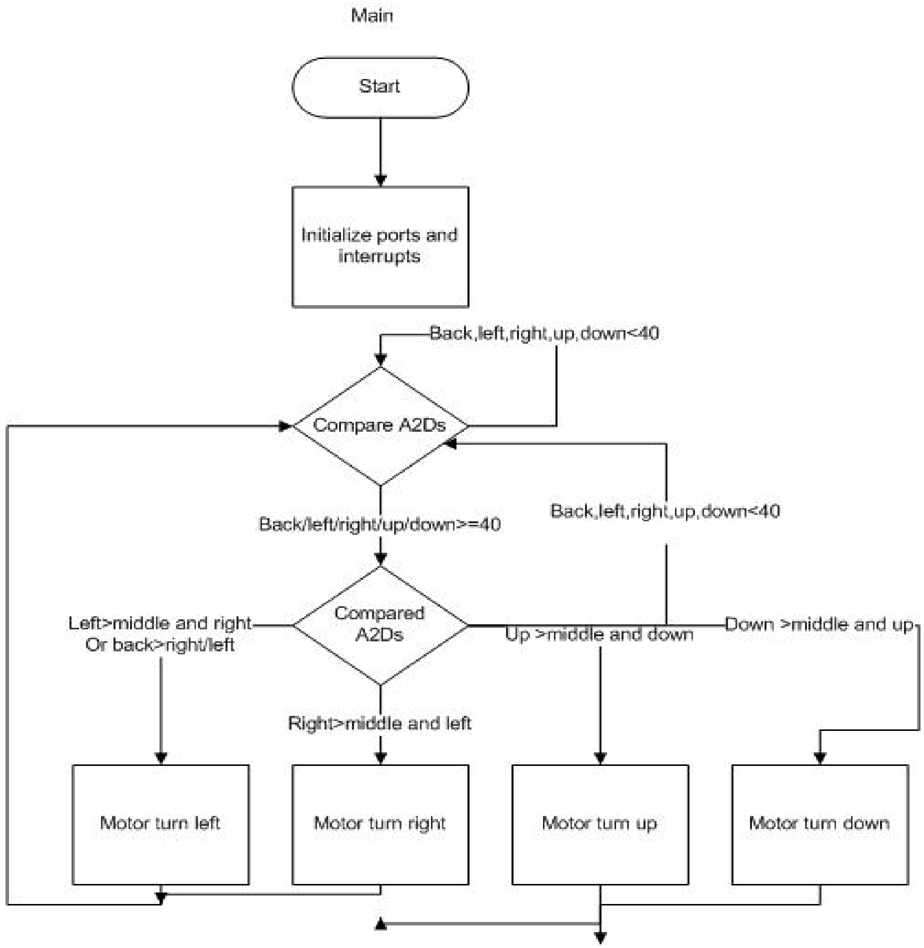


Figure 46: Solar Tracker Software Flow Chart

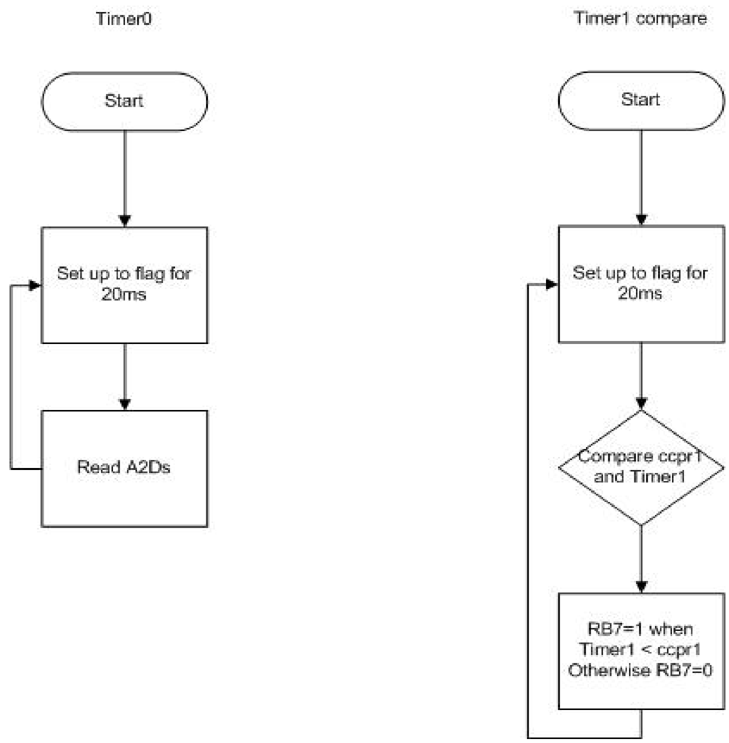


Figure 47: Solar Tracker Software Flow Chart for Timers

The Solar Tracker Software is located in the Appendix.

Heat Exchanger

Figure 48 and 49 displays the LabVIEW front panel and block diagram respectively. The front panel is the user interface, and the block diagram is the code. The following descriptions describe the operation of the front panel. A short description is also provided on figure 48.

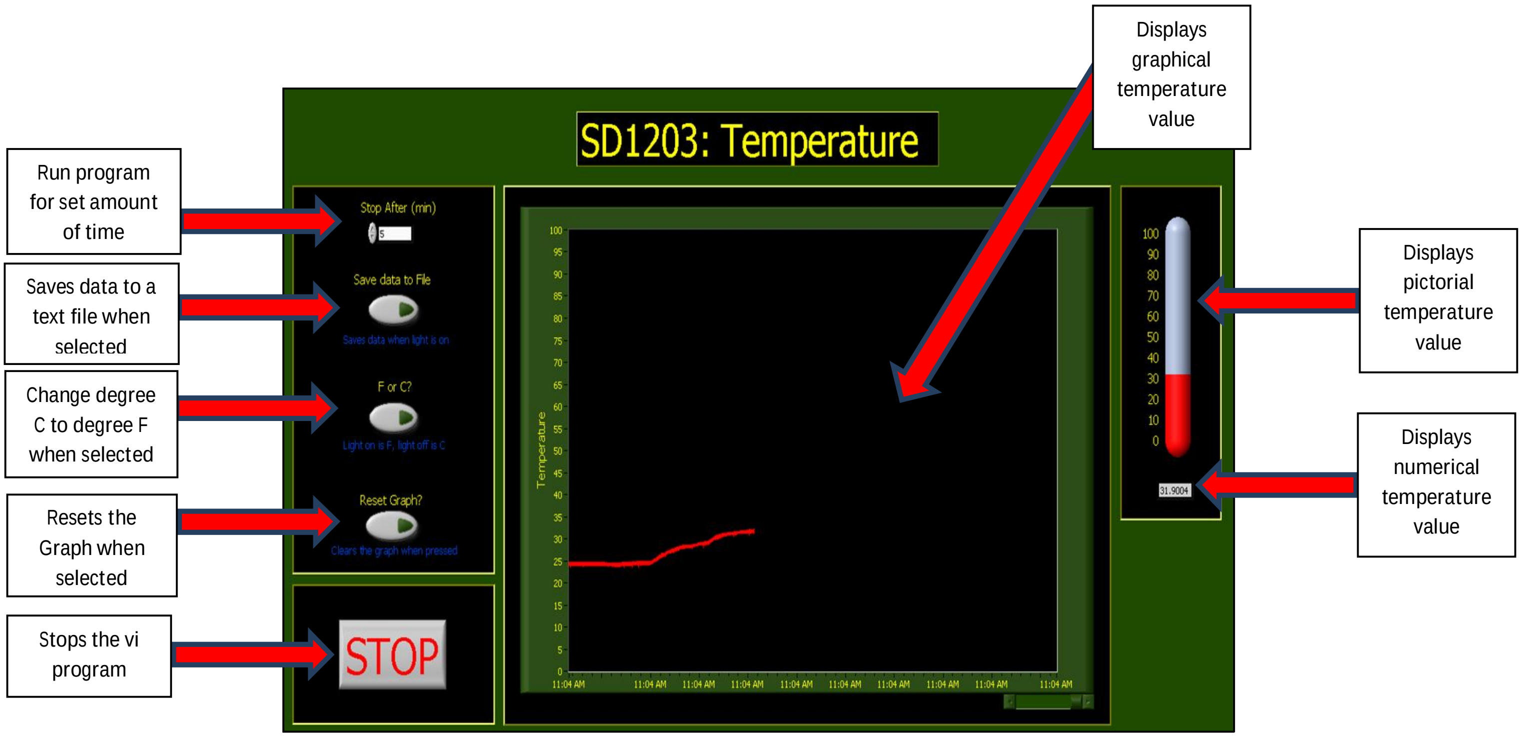


Figure 48: LabVIEW Front Panel

Stop After (min): The "Stop After (min)" control is a program timer that executes for the duration set by the user. For example, if the user enters "10" the program will execute for 10 minutes. When the program is initially opened the default "Stop After (min)" is set to zero, so this must be updated before the simulation begins.

Save data to file: The "Save data to file" button allows the user to save the current data being generated to a tab delimited text file. This file can then be saved and opened in Excel. This button is similar to an on and off light switch. When the button has not been pressed the green light is off, and when the button is pressed the green light is on and data will subsequently be saved. Data can be saved at any time while the program is running. However, the data is saved to only one file. For example, if you press the button to save data this data will be saved to the file. If the button is pressed again to turn off the saving data option and then pressed again the new data will overwrite your previous data. So if you want to save data from multiple runs you have to "save as" the data file to a new name. When the program is initially opened the default "Save data to file" is off, so if you want to save the data to the file you must select the button (the green light will turn on).

F or C?: The "F or C?" button changes the temperature reading from degrees Celsius to degrees Fahrenheit. This button is similar to a on and off light switch. When the program is initially opened the default "F or C?" is such that degrees Celsius is displayed, so if you want to display degrees Fahrenheit you must select the button (the green light will turn on).

Reset Graph?: The "Reset Graph?" button resets the graph when the button is pressed. This button resets to the off state (green light off) once the graph is cleared. Thus, the graph can be cleared multiple times during a run. When the program is initially opened the default "Reset Graph?" is off, so if you want to clear the graph you must select the button (the green light will turn on), once the graph is cleared the green light will turn off.

Figure 49 displays the LabVIEW block diagram merely for documentation and illustration.

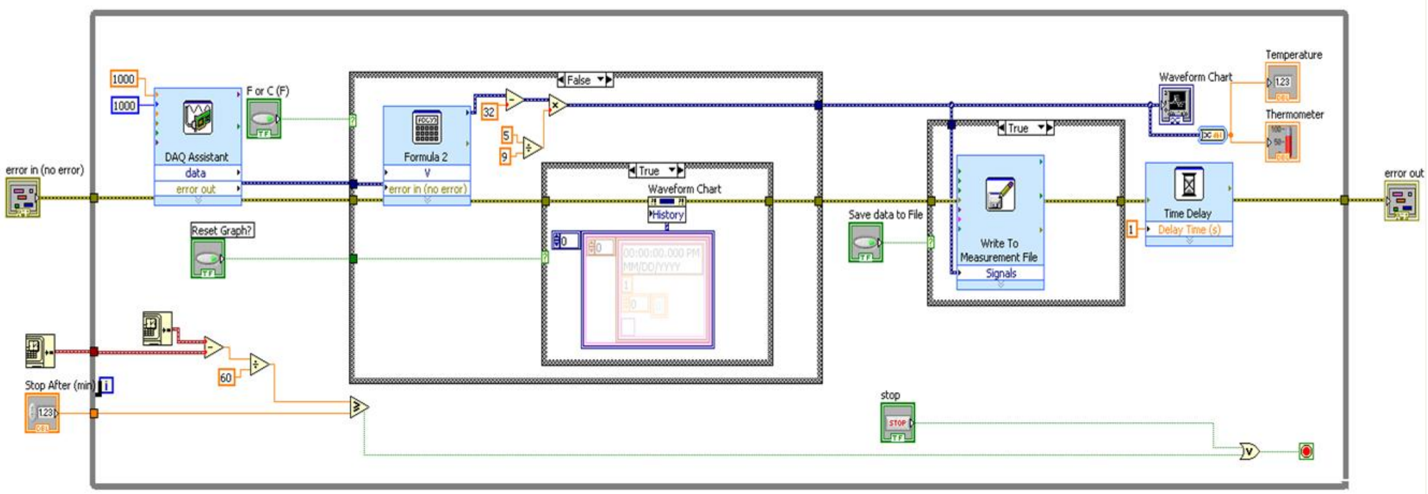


Figure 49: LabVIEW Block Diagram

Sun Simulator

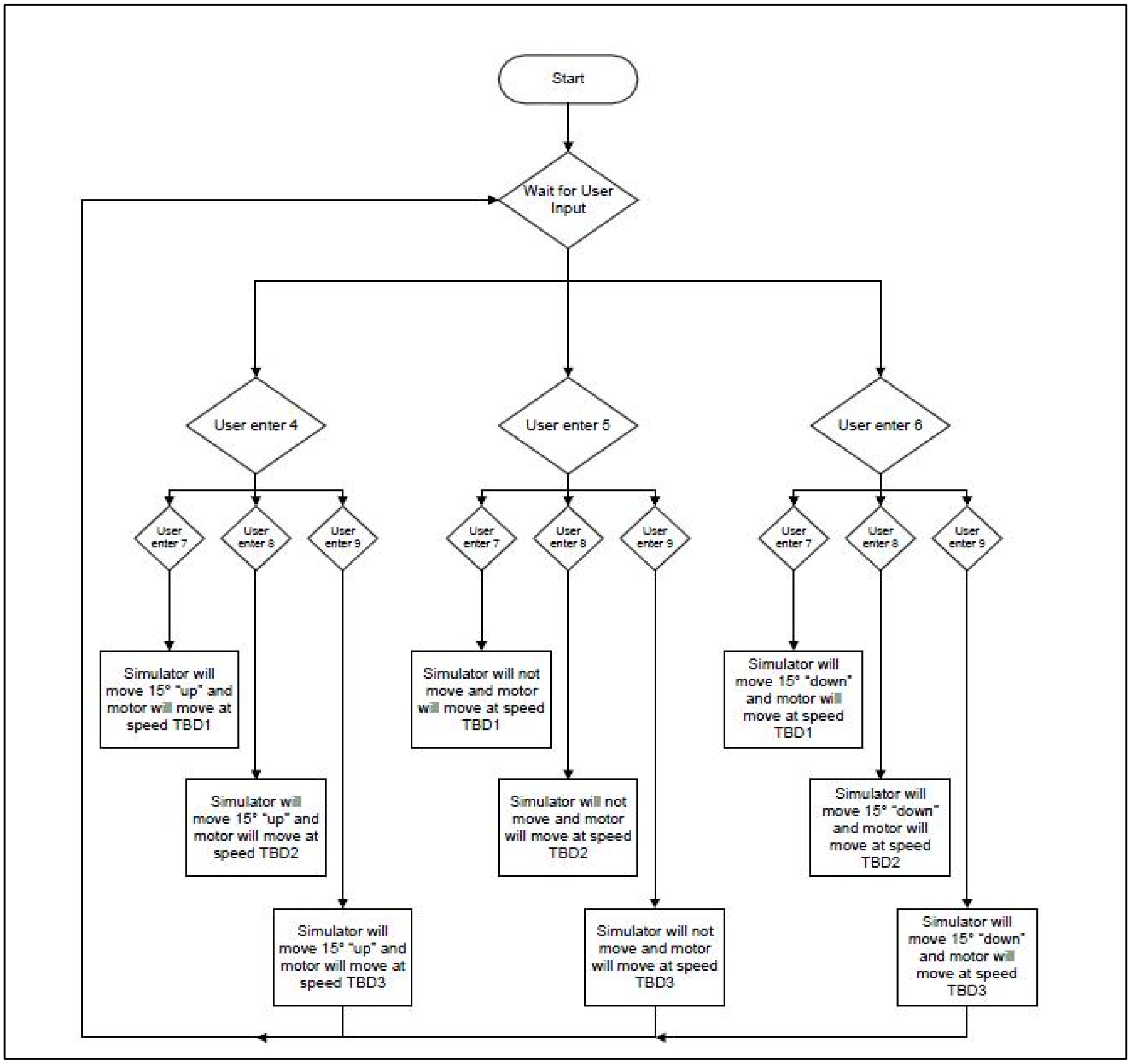


Figure 50: Sun Simulator Flow Chart

The Sun Simulator Software is located in the Appendix.

Dimmer Switch

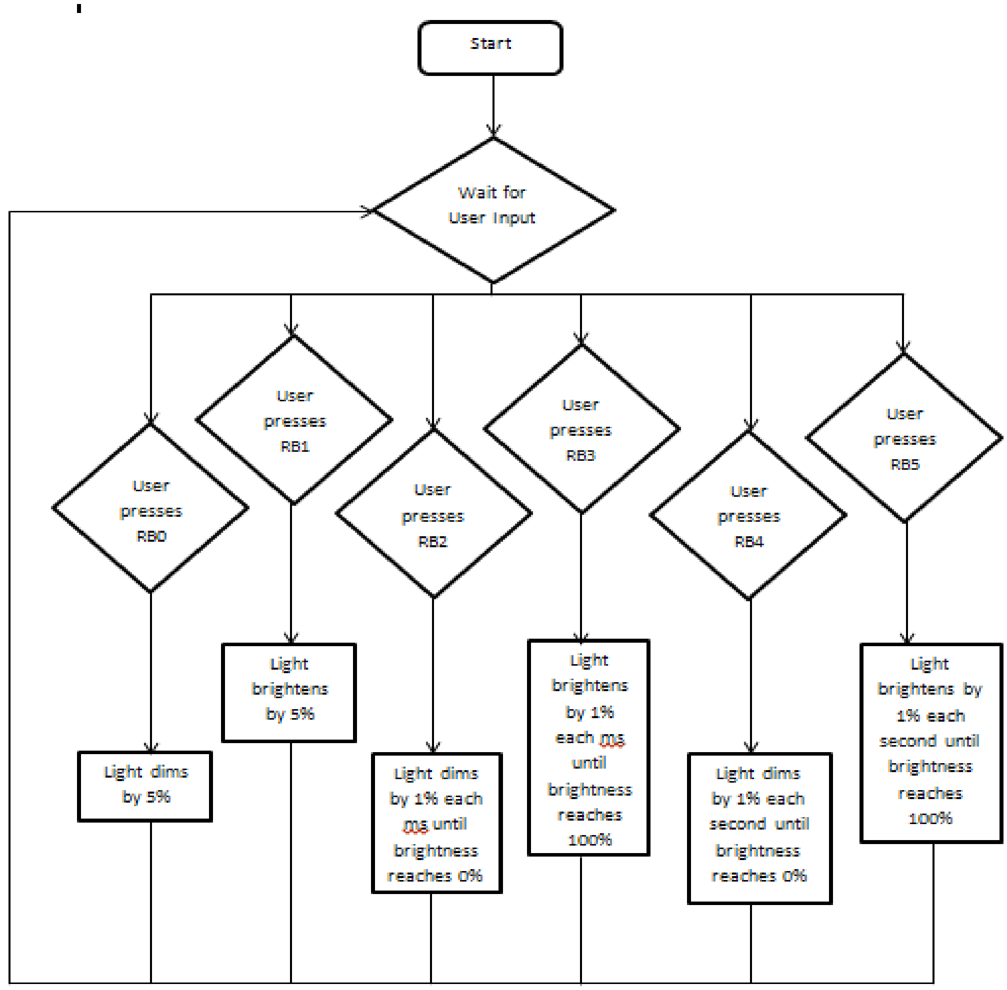


Figure 51: Dimmer Switch Flow Chart

The Dimmer Switch Software is located in the Appendix.

Technician Troubleshooting

Solar Tracker

Problem: Solar tracker does not do anything when turned on.
Solution: Check battery if power indicator does not light up when power switch is on. Otherwise check loose connection motor wire connections.

Problem: Solar tracker makes random movements
Solution: Check and make sure solar sensor is not covered or blocked from sunlight. Check solar sensor to make sure no loose wire connections or broken enclosure.

Heat Exchanger

Before using the heat exchanger verify all hardware and software is connected properly.

If the LabVIEW program is not graphing the temperature or is simply graphing a horizontal line regardless of an expected change in temperature:

1. Verify the toggle switch on the heat exchanger PCB enclosure is switched "on" and the green LED light is illuminated. If the green LED does not come on jump to #3, #4 then #5. If the green LED light is on but data is still not being graphed,
2. Verify the program is running by making sure the run arrow changes from  to , after the run button  is pressed. Verify a value was entered in the "Wait After (min)" control on the LabVIEW Front Panel. Upon starting the program the default value is zero, so the program will stop immediately and the run button will automatically return to .
3. Verify the DAQ6008 is properly connected to the computer via USB port. It may help to remove and re-connect the DAQ USB to the USB port. The DAQ will take some time to load, so be patient. After some time, if this does not solve the problem,
4. Verify the wires connecting the heat exchanger circuit to the DAQ6008 are properly connected. Verify the ground wire (black wire) is connected to port 1, the +5V wire (red wire) is connected to port 31, and the analog input wire (blue wire) is connected to port 2. Please refer to figure 52 for the DAQ wiring diagram. NOTE: If a wire appears to be connected incorrectly unplug the DAQ from the USB port immediately before attempting to reconnect the wire. If a wire is not connected, unplugged the DAQ USB form the USB port before connecting wires (or you can remove the power or ground wires) to prevent damage to the heat exchanger PCB or to the DAQ. If this does not solve the problem and the heat exchanger software is still not operational,

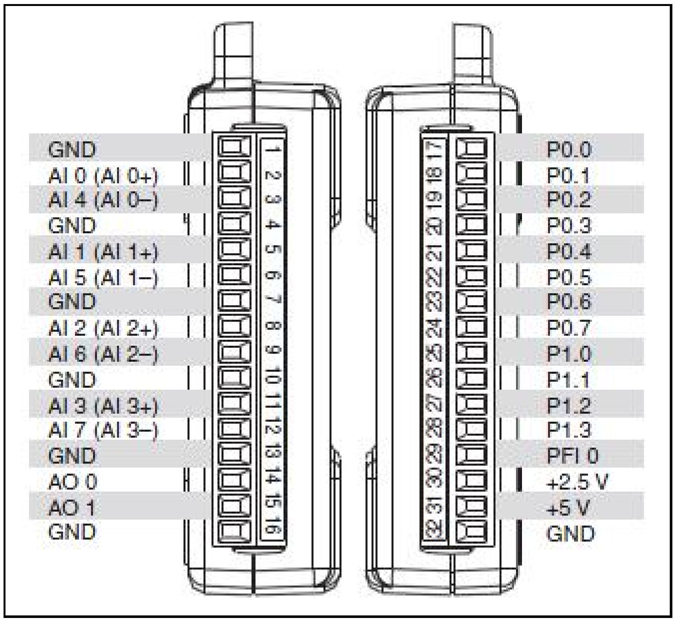


Figure 52: NI USB DAQ6008 Pinout

5. There may be a bad wire connection on the PCB. A qualified individual familiar with trouble shooting circuits will need to use a multimeter to check all connections. If a bad connection is found then re-solider the wire to the PCB trace.

Battery Charger

Insert troubleshooting information here.

Sun Simulator

If the Sun Simulator is not functional when an option is pressed on the keyboard:

1. Verify the power supply is turned on. First, when the power is not turned on remove the red and black power cables on the power supply (these are the wire providing power to the Sun Simulator). Now turn on the power source by pressing the green power button. When the power is “on” verify the voltage level is set to +5V. When the power supply voltage is not set to +5V, turn the voltage knob to set the power supply to +5V. Once the power supply is set to +5V turn the power supply off and re-connect the red and black cable to the power supply. Once connected turn the power supply back “on” and now the Sun Simulator is ready to simulate. NOTE: The Sun Simulator PCB can only run with +5V, voltages higher than +5V may damage the PCB. If this does not solve the problem,
2. Verify the leads from the power supply are connected to the black (ground) and red (+5V) connectors on the PCB enclosure. These connectors transfer the power to the Sun Simulator PCB. If the wires are not connected, connect them when the power is off. If the wire are connected,

3. With the power off, verify all wire connections a securely connected to the PCB enclosure. These wires include the power wires mentioned in #2, the wires from the stepper motor, and the wires from the linear actuator. If there are wires not securely connected, connect them with the power off. If all wires are securely connected to the PCB enclosure and there is still a problem,
4. Remove the PCB enclosure top and verify the wire connections from #3 are securely connected from the PCB enclosure to the PCB. If there are wires not securely connected, connect them with the power off. If all wires are securely connected to the PCB and there are still problems,
5. Verify the wires coming from the keypad are securely connected to the PCB. If there are wires not securely connected, connect them with the power off. If all wires are securely connected to the PCB enclosure and the Sun Simulator is still not operational,
6. There may be a bad wire connection on the PCB. A qualified individual familiar with trouble shooting circuit will need to use a multimeter to check all connections. If a bad connection is found then re-solider the wire to the PCB trace.

Dimmer Switch

Before using the dimmer switch, ensure that all hardware is connected properly.

Problem: When the dimmer switch is turned on, the light does not turn on.
Solution: Press the yellow reset button and hold for a couple of seconds. Then ensure that the dimmer is not set to "PWM = 1.00". This would indicate that the light is 100% off. This light level can be adjusted manually to any level by pressing RB0 or RB1. NOTE: The light is fully off when the display reads "PWM = 1.00," and the light is fully on when the display reads "PWM = 0.00".

Problem: The dimmer function is not responding to any buttons.
Solution: Check and make sure the power source is plugged into the dimmer PCB correctly. Also, verify that the light is plugged into the dimmer PCB correctly.

Project Comments

The project was fun, challenging, and time consuming. It took a lot of research and planning, especially during design II and the beginning of design III, to accomplish what was completed. A large part of the project required mechanical design and construction, which proved to be the most challenging. The mechanical portions of the project made the overall project more exiting, and being able to see a completed project is something to be proud of.

Solar Tracker Future Improvements

The Satellite System was never fully constructed and tested. We attempted to constructed the system but found that when the tubing was connected to the coils, heat exchanger, and reservoir, the tubing put a lot of strain on the Solar Tracker motor used to rotate the base. So, one improvement is a larger

motor that can produce more torque, so the base can be rotated when the tubing is attached, will most likely have to be installed. Also, having longer tubing will also help. In addition, a stronger pump in the reservoir may have to be used to help circulate the water.

Sun Simulator Issues

The current design utilizes four stationary pulleys and two sheave pulleys (for rolling the lamp on the track). The stationary pulleys are used to control and "track" the wire pulling the lamp. Before the completed simulator, the force from the geared stepper motor to pull the sheave pulley back and forth was completely horizontal. This horizontal force put a large strain on the track and did not move the lamp. The geared stepper motor (which Jeff made from a washing machine gear box) has enough torque to pull the sheave pulleys, however the force is basically being applied horizontally onto the track. What I did to correct this was to add a wire support so the force on the wire is no longer horizontal but angular. The wire support worked great and allowed the sheave pulleys to move smoothly along the track. The initial design had the lamp being hung from the sheave pulleys, but because of the wire support I had to connect the lamp on top of the sheave pulleys. This did not cause any new problems; in fact I think it is a better design.

Another problem I encountered was with the linear actuator. A ball and socket elbow joint is being used to relieve the tension caused when the track is being tilted, without this elbow joint the motor would bind to stop. The joint worked great and provided full tilt in both directions (with some slipping). One day the joint "stiffened" and I could not figure out why. Because of this the motor seized when the track was at a certain angle. As I did not know why the joint stiffened I tried lubrication to fix the problem, which helped but did not solve the problem. I eventually found out the joint stiffened because the bolt I used to secure the elbow was torqued to hard into the dome of the elbow joint, piecing the dome. To fix the problem I had to order another ball joint (fortunately they only cost around $3). Before installing the new elbow joint I grinded off the bolt tip, so I would not have the same problem. The new elbow joint helped a lot but the motor still slips at certain angles. This is not a hardware issue but a mechanical design issue.

Lastly, I decided to use a push button keypad because I was having reliability problems with the black keypads. I had working code at the end of the spring semester (end of Design II), however when I tried the code again in the fall the keypad worked only part of the time. I then changed the code and finally got the code working again (with new code); a few days later the keypad only worked part of the time. At this point I decided it would it be bad if the keypad did not work after ordering the PCB and especially if it decided to not work on Demo day. The pushbuttons were always reliable, so I made a keypad out of pushbuttons. Although this worked with the code, constructing and wiring the keypad was very time consuming. However, the keypad works and Sun Simulator successfully functioned on Demo day.

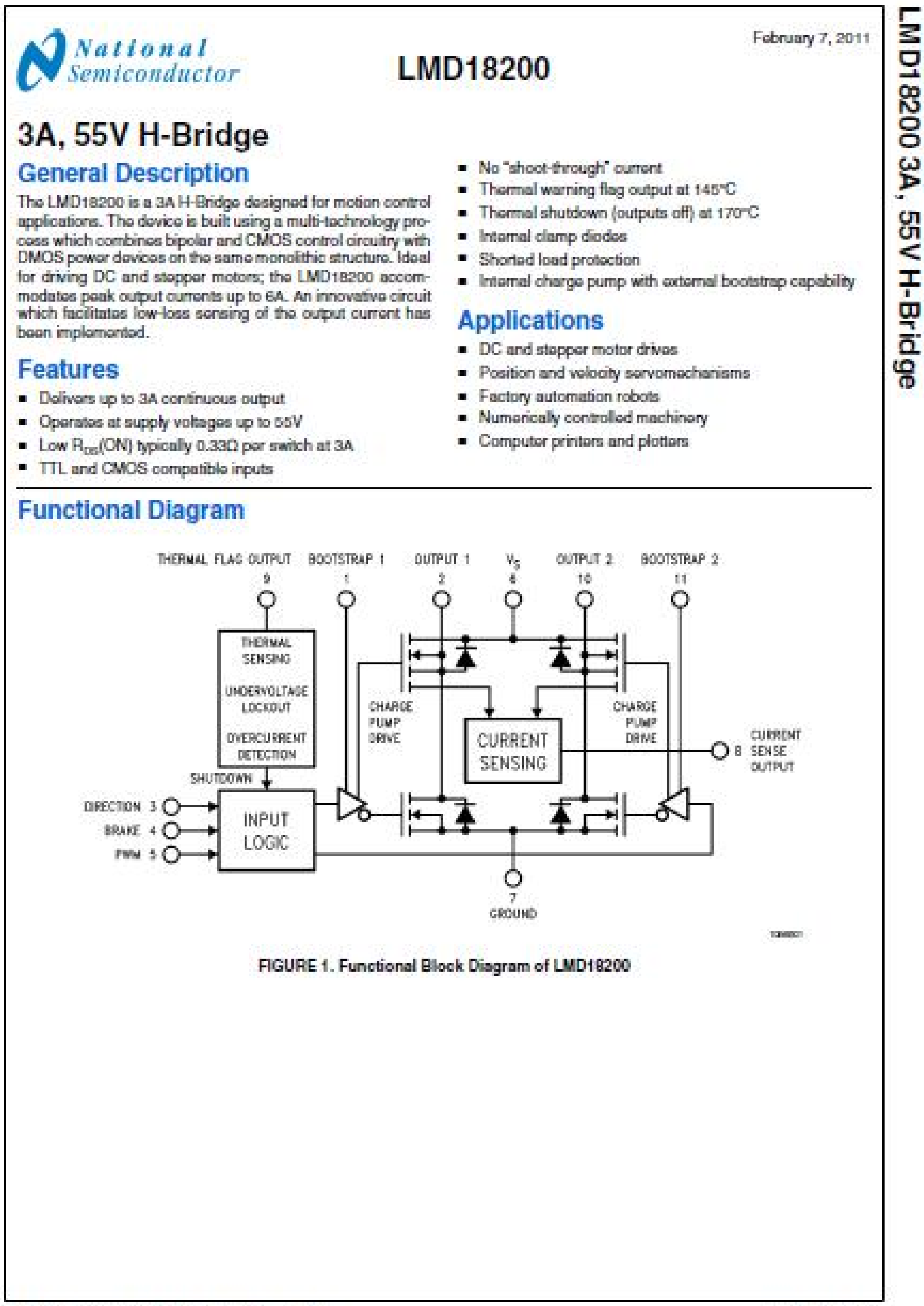
Summary

The goal of this project was to create a functional solar tracking system that is operational outside using natural sunlight. With the Mylar reflective material, the reflected sun energy will heat liquid running through a copper coil connected to a water reservoir. Temperature data will be plotted real time using LabVIEW, so decrease in temperature due to cloud cover should be easily recognizable. When the solar panel is installed, using the energy obtained a buck/boost battery charger will charge one 12-volt rechargeable battery. A second battery (already charged) will power a 300-watt inverter, which will allow for use of household appliances.

Appendix

Solar Tracker

Solar Tracker Data Sheets



National Semiconductor

LMD18200

February 7, 2011

LMD18200 3A, 55V H-Bridge

3A, 55V H-Bridge

General Description

The LMD18200 is a 3A H-Bridge designed for motion control applications. The device is built using a multi-technology process which combines bipolar and CMOS control circuitry with DMOS power devices on the same monolithic structure. Ideal for driving DC and stepper motors; the LMD18200 accommodates peak output currents up to 6A. An innovative circuit which facilitates low-loss sensing of the output current has been implemented.

Features

- Delivers up to 3A continuous output
- Operates at supply voltages up to 55V
- Low R_{DS}(ON) typically 0.33Ω per switch at 3A
- TTL and CMOS compatible inputs
- No "shoot-through" current
- Thermal warning flag output at 145°C
- Thermal shutdown (outputs off) at 170°C
- Internal clamp diodes
- Shorted load protection
- Internal charge pump with external bootstrap capability

Applications

- DC and stepper motor drives
- Position and velocity servomechanisms
- Factory automation robots
- Numerically controlled machinery
- Computer printers and plotters

Functional Diagram

FIGURE 1. Functional Block Diagram of LMD18200

Figure 53: H-Bridge Data Sheet

Solar Tracker Software

```
// Solar tracking code final
//Author: Qingyu Meng (Jace)

//h bridge config: digit0: direction, digit1: breaking.
// PORTC x axis motor, PORTD Y axis motor.

#include    <pic18.h>
#include      "lcd_portd.h"
#include      "function.h"
#include      "function.c"
#include      "LCD_PortD.C"

void A2D_Init(void)
{
              TRISA = 0xFF;
              TRISE = 0x0F;
              ADCON2 = 0x85;
              ADCON1 = 0x07;
              ADCON0 = 0x01;
}

unsigned int A2D_Read(unsigned char c)
{
  unsigned int result;
  unsigned char i;
  c = c & 0x0F;
  ADCON0 = (c << 2) + 0x01; // set Channel Select
  for (i=0; i<10; i++); // wait 2.4us (approx)
  GODONE = 1; // start the A/D conversion
  while(GODONE); // wait until complete (approx 8us)
  return(ADRES);
 }

unsigned int back,left,right,mid,up,down;
unsigned int thold = 40; // to be determined

void interrupt IntServe(void)
{
       if(TMR0IF)
       {
              TMR0 = -50000;                    //every 20ms

              back = A2D_Read(0);               // Read A2D
              left = A2D_Read(1);
              right = A2D_Read(2);
              mid = A2D_Read(3);
              up = A2D_Read(4);
              down = A2D_Read(5);

              TMR0IF = 0;
       }

       if(TMR1IF) {
              TMR1 = -50000;                    // flag every 20 ms
              RB7 = 1;                          // output to RD2
              TMR1IF = 0;
       }
       if(CCP1IF) {                             //PWM
              RB7 = 0;
              CCP1IF = 0;
       }
}
```

```
void main(void)   // main function
{
        unsigned int PWM = 37500; // 30% duty cycle;
        CCPR1 = -PWM;

        //Initialize Timer 0
        T0CON = 0x80;                           //PS = 2
        TMR0 = -50000;                          //flag every 20ms
        TMR0ON = 1;
        TMR0IE = 1;
        TMR0IP = 1;

        //Initialize Timer 1 Compare
        T1CON           = 0x91;                 // PS = 2
        TMR1            = -50000;               // 20 ms cycles
        CCP1CON         = 0x0A;
        TMR1ON          = 1;
        TMR1IE          = 1;
        TMR1IP          = 1;
        CCP1IE          = 1;

         //Initialize ports
        TRISB = 0x7F; //for testing motors
        TRISC = 0;   //x axis motor controller
        TRISD = 0;   // y axis motor controller
        PORTC = 0x2; //motor breaking
        PORTD = 0x2; //motor breaking

        //Enable all Interrupts
        GIE    = 1;
        PEIE  = 1;

        A2D_Init();

        while(1)
        {
                if (RB0||RB1||RB2||RB3)         // manual testing motors
                {                               // manual testing motors in 4 directions:

                        while (RB0)
                                PORTC = 0;

                        while (RB1)
                                PORTC = 0x1;

                        while (RB2)
                                PORTD = 0;

                        while (RB3)
                                PORTD = 0x1;

                                PORTC = 0x2;            //motor breaking
                                PORTD = 0x2;            //motor breaking
                }

                        while (back>thold || left>thold || right>thold || up>thold || down>thold)// check whether it is too dark outside
                        {

                        if (RB0||RB1||RB2||RB3)                 // manual testing motors
                        break;

                        while(back>=left && back>=right) // checking whether panel front is facing the sun
```

```
{
    PORTC = 0; // turning left if facing back
    if (RB0||RB1||RB2||RB3)         // manual testing motors
    break;
    }
        while(back<left || back<right)          // facing front
        {
            if (RB0||RB1||RB2||RB3)         // manual testing motors
            break;
                //x axis motor sensing
                while(left>right && left>mid)           // left side is brighter
                {
                    PORTC = 0; // turning left
                    if (RB0||RB1||RB2||RB3)         // manual testing motors
                    break;
                }
                PORTC = 0x2; //motor breaking

                //y axis motor sensing
                while(up>down && up>mid)            // upper side is brighter
                {
                    PORTD = 0; // turning up
                    if (RB0||RB1||RB2||RB3)          // manual testing motors
                    break;
                }
                PORTD = 0x2; //motor breaking

                //x axis motor sensing
                while(right>left && right>mid)          // right side is brighter
                {
                    PORTC = 0x1;                    // turning right
                    if (RB0||RB1||RB2||RB3)         // manual testing motors
                    break;
                }
                PORTC = 0x2;                        //motor breaking

                //y axis motor sensing
                while(down>up && down>mid)          // down side is brighter
                {
                    PORTD = 0x1;                    // turning down
                    if (RB0||RB1||RB2||RB3)         // manual testing motors
                    break;
                }
                PORTD = 0x2;                        // motor breaking

        }

    }
  }
}
```

Heat Exchanger Data Sheets

Part Numbering

NTC Thermistors for Temp. Sensor and Compensation Chip Type

(Part Number) NC P 18 XH 103 J 03 RB

Product ID

Product ID	
NC	NTC Thermistors Chip Type

Series

Code	Series
P	Plated Termination Series

Dimensions (L×W)

Code	Dimensions (L×W)	EIA
03	0.60×0.30mm	0201
15	1.00×0.50mm	0402
18	1.60×0.80mm	0603
21	2.00×1.25mm	0805

Temperature Characteristics

Code	Temperature Characteristics
XC	Nominal B-Constant 3100–3149K
XF	Nominal B-Constant 3250–3299K
XH	Nominal B-Constant 3350–3399K
XM	Nominal B-Constant 3500–3549K
XQ	Nominal B-Constant 3650–3699K
XV	Nominal B-Constant 3900–3949K
XW	Nominal B-Constant 3950–3999K
WB	Nominal B-Constant 4050–4099K
WD	Nominal B-Constant 4150–4199K
WF	Nominal B-Constant 4250–4299K
WL	Nominal B-Constant 4450–4499K
WM	Nominal B-Constant 4500–4549K

Resistance

Expressed by three-digit alphanumerics. The unit is ohm (Ω).
The first and second figures are significant digits, and the third figure expresses the number of zeros which follow the two figures.

Ex.

Code	Resistance
102	1kΩ
103	10kΩ
104	100kΩ

Resistance Tolerance

Code	Resistance Tolerance
D	±0.5%
F	±1%
E	±3%
J	±5%

Individual Specifications

Structures and others are expressed by two figures.

Ex.

Code	Individual Specifications
03	Standard Type

Packaging

Code	Packaging
RA	Plastic Taping 4mm Pitch (4000 pcs.)
RB	Paper Taping 4mm Pitch (4000 pcs.)
RC	Paper Taping 2mm Pitch (10000 pcs.)
RL	Paper Taping 2mm Pitch (15000 pcs.)

2

muRata

Figure 54: NTC Thermistor Data Sheet

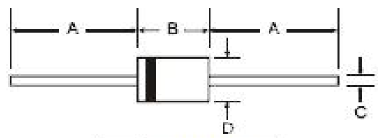
Data Sheets

1N4001 - 1N4007

1.0A RECTIFIER

Features

- Diffused Junction
- High Current Capability and Low Forward Voltage Drop
- Surge Overload Rating to 30A Peak
- Low Reverse Leakage Current
- **Lead Free Finish, RoHS Compliant (Note 3)**



Dim	DO-41 Plastic Min	Max
A	25.40	—
B	4.06	5.21
C	0.71	0.864
D	2.00	2.72
All Dimensions in mm		

Mechanical Data

- Case: DO-41
- Case Material: Molded Plastic. UL Flammability Classification Rating 94V-0
- Moisture Sensitivity: Level 1 per J-STD-020D
- Terminals: Finish - Bright Tin. Plated Leads Solderable per MIL-STD-202, Method 208
- Polarity: Cathode Band
- Mounting Position: Any
- Ordering Information: See Page 2
- Marking: Type Number
- Weight: 0.30 grams (approximate)

Maximum Ratings and Electrical Characteristics @T_A = 25°C unless otherwise specified

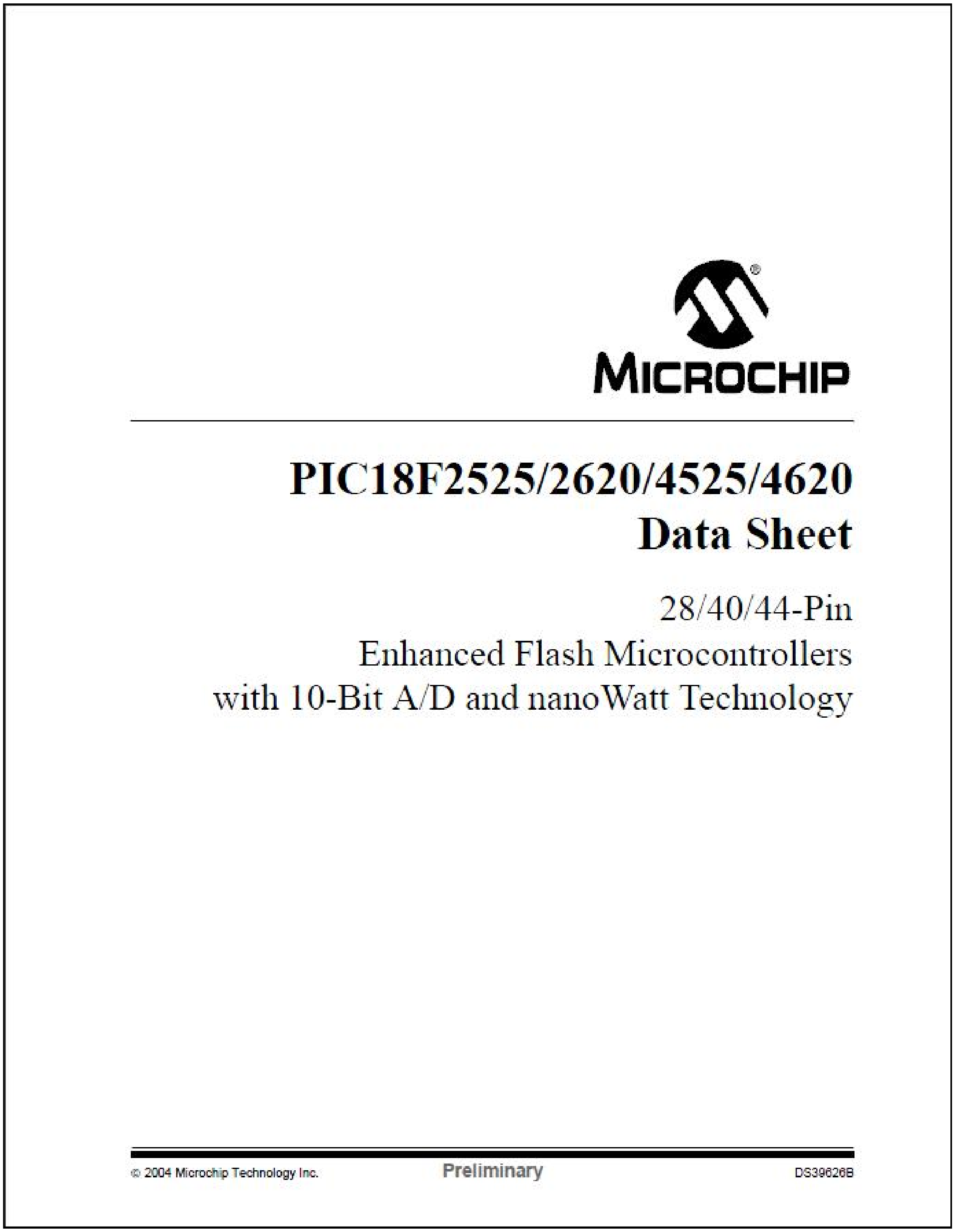
Single phase, half wave, 60Hz, resistive or inductive load.
For capacitive load, derate current by 20%.

Characteristic	Symbol	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	Unit
Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage	V_{RRM} V_{RWM} V_R	50	100	200	400	600	800	1000	V
RMS Reverse Voltage	$V_{R(RMS)}$	35	70	140	280	420	560	700	V
Average Rectified Output Current (Note 1) @ T_A = 75°C	I_O	1.0							A
Non-Repetitive Peak Forward Surge Current 8.3ms single half sine-wave superimposed on rated load	I_{FSM}	30							A
Forward Voltage @ I_F = 1.0A	V_{FM}	1.0							V
Peak Reverse Current @T_A = 25°C at Rated DC Blocking Voltage @ T_A = 100°C	I_{RM}	5.0 50							μA
Typical Junction Capacitance (Note 2)	C_j	15				8			pF
Typical Thermal Resistance Junction to Ambient	$R_{\theta JA}$	100							K/W
Maximum DC Blocking Voltage Temperature	T_A	+150							°C
Operating and Storage Temperature Range	T_J, T_{STG}	-65 to +150							°C

Notes:
1. Leads maintained at ambient temperature at a distance of 9.5mm from the case.
2. Measured at 1.0 MHz and applied reverse voltage of 4.0V DC.
3. EU Directive 2002/95/EC (RoHS). All applicable RoHS exemptions applied, see EU Directive 2002/95/EC Annex Notes.

DS28002 Rev. 8 - 2 1 of 3 www.diodes.com 1N4001-1N4007 © Diodes Incorporated

Figure 55: 1N4002 Diode Data Sheet



MICROCHIP

PIC18F2525/2620/4525/4620 Data Sheet

28/40/44-Pin
Enhanced Flash Microcontrollers
with 10-Bit A/D and nanoWatt Technology

© 2004 Microchip Technology Inc. Preliminary DS39626B

Figure 56: PIC18F4620 Data Sheet

NPN SILICON PLANAR MEDIUM POWER HIGH GAIN TRANSISTOR

ZTX1051A

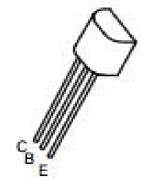
ISSUE 3 – FEBRUARY 95

FEATURES

* B_{CEV}=150V
* Very Low Saturation Voltage
* High Gain
* Inherently Low Noise

APPLICATIONS

* Emergency Lighting
* Low Noise Audio



E-Line
TO92 Compatible

ABSOLUTE MAXIMUM RATINGS.

PARAMETER	SYMBOL	VALUE	UNIT
Collector-Base Voltage	V_{CBO}	150	V
Collector-Emitter Voltage	V_{CEO}	40	V
Emitter-Base Voltage	V_{EBO}	5	V
Peak Pulse Current	I_{CM}	10	A
Continuous Collector Current	I_C	4	A
Base Current	I_B	500	mA
Power Dissipation at T_{amb}=25°C	P_{tot}	1	W
Operating and Storage Temperature Range	T_j:T_{stg}	-55 to +200	°C

ZETEX

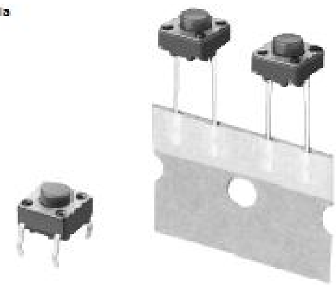
Figure 57: ZTX1051A Data Sheet

Panasonic Light Touch Switches/5N Type

5N Type Light Touch Switches

Japan
Malaysia
China

Type: **EVQPA/EVQPB/EVQPF**
EVQ2/EVQPC



Broad product range meets a variety of needs

■ Features

- 6.0 mm×6.0 mm square, body thickness 3.2 mm
- Travel 0.25 mm (clear click feeling)
- Wide selection with or without ground terminal, top-push or side-operation type with snap-in terminals available
- Wave soldering applicable

■ Recommended Applications

- Operation button switches for audio/visual, OA, communication equipment, measurement instruments

■ Explanation of Part Numbers

1	2	3	4	5	6	7	8	9
E	V	Q						

Height (7th–8th), Push Plate Color (9th)

4th, 5th	Type
PA, PB PF, 2□ PC	5N Type

■ Product Chart

○ = In production

Operating Force	Positioning Pin	Ground Terminal	Top-push Type				Side-operation Type			
			4.3 mm	5.0 mm	7.0 mm	9.5 mm	3.15 mm	3.85 mm	5.85 mm	8.35 mm
1.0 N	Without positioning pin	With ground terminal	○	○	○	○	○	○	○	○
		Without ground terminal	○	○	○	○	—	—	—	—
1.3 N	Without positioning pin	With ground terminal	○	○	○	○	○	○	○	○
		Without ground terminal	○	○	○	○	—	—	—	—
1.6 N	Without positioning pin	With ground terminal	○	○	○	○	○	○	○	○
		Without ground terminal	○	○	○	○	—	—	—	—
2.6 N	Without positioning pin	With ground terminal	○	○	○	○	○	○	○	○
		Without ground terminal	○	○	○	○	—	—	—	—

Design and specifications are each subject to change without notice. Ask factory for the current technical specifications before purchase and/or use.
Should a safety concern arise regarding this product, please be sure to contact us immediately.

Sep. 2005

Figure 58: Panasonic EVQ Push Button Data Sheet

Sun Simulator Software

The following is the code for the Sun Simulator motor controller. The software is programmed in C Programming and built and compiled in MPLAB IDE v8.73.

```
//	SD1203 Sun Simulator
//	OVERVIEW: This code controls the stepper motor and linear actuator for
//			for simulations.  The heat lamp has 3 speed controls and the
//			track has 3 angle settings.
//
//			ROW 1: lamp speed 1
//			ROW 2: lamp speed 2
//			ROW 3: lamp spped 3
//			Column 1: track angle 1
//			Column 2: track angle 2
//			Column 3: track angle 3
//
//------------------------------------------------------------------------------------------
// Revision History
//
// DATE	INITIALS  DESCRIPTION
// 2/27/12		BMA	Initial Design of motor operation
// 3/07/12		BMA	Added keypad operation to select motor speed
// 3/09/12		BMA	Added second motor for simulator angle
// 3/25/12		BMA	Made code adjustments for simultaneous motor operation
// 3/26/12		BMA	Made further code adjustments for simultaneous motor operation
// 4/25/12		BMA	Updated keypad code and motor speeds for demonstration day
// 9/13/12		BMA	Rewrote the code for the keypad
// 9/27/12		BMA	Made code updates, Jace helped with keypad and motor code
// 10/2/12		BMA	Removed keypad code, using push buttons for a keypad, code was updated
// 11/6/12		BMA	Updated motor operation to correlate with simulator (simulator is built)
// 11/10/12		BMA	Finalized code, ready for demo day
//------------------------------------------------------------------------------------------

// Global Variables
unsigned int i = 0;
unsigned int a = 0;
unsigned int N = 0;
unsigned int speed = 0;
const unsigned char STEP[4] = {1,2,4,8};
int TEMP=0, X;

// Subroutine Declarations
#include <htc.h>
#include <pic18.h>
#include "FUNCTION.h";

// Subroutines
#include "FUNCTION.c";
#include "Lcd_20x4.c";

// Functions
void Stepper1(char spin,unsigned int steps){
		if(spin == 0){		//Forward for 'step' steps
				for (i=0; i<steps; i++){
```

```
                N = i % 4;
                PORTD = STEP[N];
                Wait_ms(speed);
                }
        }
    else if(spin == 1){
        for(i = steps; i>0; i--){
                N = i % 4;
                PORTD = STEP[N];
                Wait_ms(speed);
                }
        }
}

void Stepper2(char spin,unsigned int steps){
     if(spin == 0){         //Forward for 'step' steps
         for (i=0; i<steps; i++){
                N = i % 4;
                PORTA = STEP[N];
                Wait_ms(speed);
                }
        }
    else if(spin == 1){
        for(i = steps; i>0; i--){
                N = i % 4;
                PORTA = STEP[N];
                Wait_ms(speed);
                }
        }
}

// Main Routine
void main(void)
{
    unsigned int i, j, k;

    TRISA = 0;
    TRISB = 0xFF;
    TRISC = 0xFF;
    TRISD = 0;
    TRISE = 0;
    ADCON1 = 15;

    while(1) {
        PORTB = 0;
        PORTA = 0;
        PORTD = 0;

        int m=5000;
        while(RB1==1)
        {
                m--;
        PORTA=STEP[m%4];
                Wait_ms(10);
        }
```

```
while(RB2==1)
{
		m++;
PORTA=STEP[m%4];
		Wait_ms(10);
}

while(RB3==1)
{
		m--;
PORTD=STEP[m%4];
		Wait_ms(10);
}

while(RB4==1)
{
		m++;
		PORTD=STEP[m%4];
		Wait_ms(10);
}
		if(RB5==1){				// Button_11 (10 minutes)
		for (i=0;i<388;i++){
			speed = 10;
			Stepper2(1,50);
			PORTA = 0;
			Wait_ms(650);
```

Currently the code on the MCU for button 11 is set to run the simulation for approximately 4.5 minutes (for demo day), so the code shown here will change the setting to run the simulation for 10 minutes. All other setting are the same.

```
		}
		for (j=0;j<93;j++){
			speed = 10;
			Stepper2(0,200);
			Wait_ms(50);
		}
		Wait_ms(2000);
}
else if(RB6==1){				// Button_12 (1 hour)

		for (i=0;i<93;i++){
			speed = 10;
			Stepper2(1,200);
			PORTA = 0;
			Wait_ms(23300);
		}
		for (j=0;j<93;j++){
			speed = 10;
			Stepper2(0,200);
			Wait_ms(10);
		}
		Wait_ms(2000);
}
else if(RB7==1){				// Button_13 (4 hour)
		for (i=0;i<93;i++){
			speed = 10;
```

```
			Stepper2(1,200);
			PORTA = 0;
			Wait_ms(54000);
			Wait_ms(54000);
		}
		for (j=0;j<93;j++){
			speed = 10;
			Stepper2(0,200);
			Wait_ms(10);
		}

		Wait_ms(2000);
}
else if(RC0==1){				// Button_21 (10 minutes with tilt 15 degrees back)
		for (k=0;k<17;k++){
			speed = 10;
			Stepper1(0,200);
			Wait_ms(10);
		}
		PORTD = 0;
		for (i=0;i<388;i++){
			speed = 10;
			Stepper2(1,50);
			PORTA = 0;
			Wait_ms(650);
		}
		for (j=0;j<93;j++){
			speed = 10;
			Stepper2(0,200);
			Wait_ms(50);
		}
		PORTA = 0;
		for (k=0;k<17;k++){
			speed = 10;
			Stepper1(1,200);
			Wait_ms(10);
		}
		Wait_ms(2000);
}
else if(RC1==1){				// Button_22 (1 hour with tilt 15 degrees back)
		for (k=0;k<17;k++){
			speed = 10;
			Stepper1(0,200);
			Wait_ms(10);
		}
		PORTD = 0;
		for (i=0;i<93;i++){
			speed = 10;
			Stepper2(1,200);
			PORTA = 0;
			Wait_ms(23300);
		}
		for (j=0;j<93;j++){
			speed = 10;
			Stepper2(0,200);
```

```
				Wait_ms(10);
			}
			PORTA = 0;
			for (k=0;k<17;k++){
				speed = 10;
				Stepper1(1,200);
				Wait_ms(10);
			}
			Wait_ms(2000);
		}
		else if(RC2==1){				// Button_23 (4 hours with tilt 15 degrees back)
			for (k=0;k<17;k++){
				speed = 10;
				Stepper1(0,200);
				Wait_ms(10);
			}
			PORTD = 0;
			for (i=0;i<93;i++){
				speed = 10;
				Stepper2(1,200);
				PORTA = 0;
				Wait_ms(54000);
				Wait_ms(54000);
			}
			for (j=0;j<93;j++){
				speed = 10;
				Stepper2(0,200);
				Wait_ms(10);
			}
			PORTA = 0;
			for (k=0;k<17;k++){
				speed = 10;
				Stepper1(1,200);
				Wait_ms(10);
			}
			Wait_ms(2000);
		}
		else if(RC3==1){				// Button_31 (10 minutes with tilt 15 degrees forward)
			for (k=0;k<17;k++){
				speed = 10;
				Stepper1(1,200);
				Wait_ms(10);
			}
			PORTD = 0;
			for (i=0;i<388;i++){
				speed = 10;
				Stepper2(1,50);
				PORTA = 0;
				Wait_ms(650);
			}
			for (j=0;j<93;j++){
				speed = 10;
				Stepper2(0,200);
				Wait_ms(10);
			}
```

```
            PORTA = 0;
            for (k=0;k<17;k++){
                    speed = 10;
                    Stepper1(0,200);
                    Wait_ms(10);
            }
            Wait_ms(2000);
    }
    else if(RC4==1){                        // Button_32 (1 hour with tilt 15 degrees forward)
            for (k=0;k<17;k++){
                    speed = 10;
                    Stepper1(1,200);
                    Wait_ms(10);
            }
            PORTD = 0;
            for (i=0;i<93;i++){
                    speed = 10;
                    Stepper2(1,200);
                    PORTA = 0;
                    Wait_ms(23300);
            }
            for (j=0;j<93;j++){
                    speed = 10;
                    Stepper2(0,200);
                    Wait_ms(10);
            }
            PORTA = 0;
            for (k=0;k<17;k++){
                    speed = 10;
                    Stepper1(0,200);
                    Wait_ms(10);
            }
            Wait_ms(2000);
    }
    else if(RC5==1){                        // Button_33 (4 hours with tilt 15 degrees forward)
            for (k=0;k<17;k++){
                    speed = 10;
                    Stepper1(1,200);
                    Wait_ms(10);
            }
            PORTD = 0;
            for (i=0;i<93;i++){
                    speed = 10;
                    Stepper2(1,200);
                    PORTA = 0;
                    Wait_ms(54000);
                    Wait_ms(54000);
            }
            for (j=0;j<93;j++){
                    speed = 10;
                    Stepper2(0,200);
                    Wait_ms(10);
            }
            PORTA = 0;
            for (k=0;k<17;k++){
```

```
                speed = 10;
                Stepper1(0,200);
                Wait_ms(10);
            }
            Wait_ms(2000);
        }
    }
}
```

Dimmer

Dimmer Data Sheets

RECTRON SEMICONDUCTOR
TECHNICAL SPECIFICATION

DB101 THRU DB107

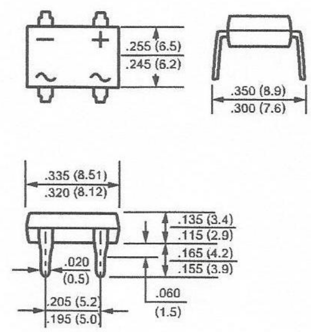
SINGLE-PHASE GLASS PASSIVATED SILICON BRIDGE RECTIFIER

VOLTAGE RANGE 50 to 1000 Volts CURRENT 1.0 Ampere

FEATURES
* Good for automation insertion
* Surge overload rating - 50 amperes peak
* Ideal for printed circuit board
* Reliable low cost construction utilizing molded
* Glass passivated device
* Polarity symbols molded on body
* Mounting position: Any
* Weight: 1.0 gram

FEATURES
* Epoxy : UL flammability classification 94V-0
* UL listed under the recognized component directory, file #E94233.

DB-1



Dimensions in inches and (millimeters)

MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS
Ratings at 25 °C ambient temperature unless otherwise specified.
Single phase, half wave, 60 Hz, resistive or inductive load.
For capacitive load, derate current by 20%.

MAXIMUM RATINGS (At TA = 25°C unless otherwise noted)

RATINGS	SYMBOL	DB101	DB102	DB103	DB104	DB105	DB106	DB107	UNITS
Maximum Recurrent Peak Reverse Voltage	V_{RRM}	50	100	200	400	600	800	1000	Volts
Maximum RMS Bridge Input Voltage	V_{RMS}	35	70	140	280	420	560	700	Volts
Maximum DC Blocking Voltage	V_{DC}	50	100	200	400	600	800	1000	Volts
Maximum Average Forward Output Current at TA = 40°C	I_O	1.0							Amps
Peak Forward Surge Current 8.3 ms single half sine-wave superimposed on rated load (JEDEC method)	I_{FSM}	50							Amps
Operating and Storage Temperature Range	T_J,T_{STG}	-65 to + 150							°C

ELECTRICAL CHARACTERISTICS (At TA = 25°C unless otherwise noted)

CHARACTERISTICS		SYMBOL	DB101	DB102	DB103	DB104	DB105	DB106	DB107	UNITS
Maximum Forward Voltage Drop per Bridge Element at 1.0A DC		V_F	1.1							Volts
Maximum Forward Voltage Drop per Bridge DC Blocking Voltage per element	@TA = 25°C	I_R	10.0							uAmps
	@TA = 125°C		0.5							mAmps

1998-8

Figure 59: H11AA1 Data Sheet

MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

Order this document by H11AA1/D

GlobalOptoisolator™

VDE | UL | CSA | SETI | SEMKO | DEMKO | NEMKO | BABT

6-Pin DIP Optoisolators AC Input/Transistor Output

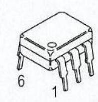
The H11AA1, H11AA2, H11AA3, H11AA4 devices consist of two gallium–arsenide infrared emitting diodes connected in inverse parallel, optically coupled to a monolithic silicon phototransistor detector.

- Built–In Protection for Reverse Polarity
- Guaranteed CTR Minimum Values as High as 100%
- Guaranteed Minimum/Maximum Symmetry Limits
- ***To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.***

H11AA1* [CTR = 20% Min]
H11AA2 [CTR = 10% Min]
H11AA3 [CTR = 50% Min]
H11AA4* [CTR = 100% Min]

*Motorola Preferred Devices

STYLE 8 PLASTIC



STANDARD THRU HOLE
CASE 730A–04

Applications

- Detecting or Monitoring ac Signals
- AC Line/Digital Logic Isolation
- Programmable Controllers
- Interfacing and coupling systems of different potentials and impedances
- AC/DC — Input Modules

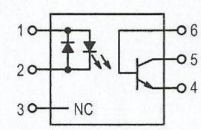
MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INPUT LED			
Forward Current — Continuous (RMS)	I_F	60	mA
LED Power Dissipation @ T_A = 25°C with Negligible Power in Output Detector Derate above 25°C	P_D	120 1.41	mW mW/°C
OUTPUT TRANSISTOR			
Collector–Emitter Voltage	V_{CEO}	30	Volts
Emitter–Base Voltage	V_{EBO}	5	Volts
Collector–Base Voltage	V_{CBO}	70	Volts
Collector Current — Continuous	I_C	150	mA
Detector Power Dissipation @ T_A = 25°C with Negligible Power in Input LEDs Derate above 25°C	P_D	150 1.76	mW mW/°C
TOTAL DEVICE			
Isolation Surge Voltage(1) (Peak ac Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac(pk)
Total Device Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	250 2.94	mW mW/°C
Ambient Operating Temperature Range(2)	T_A	–55 to +100	°C
Storage Temperature Range(2)	T_{stg}	–55 to +150	°C
Soldering Temperature (10 sec, 1/16″ from case)	T_L	260	°C

1. Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.
GlobalOptoisolator is a trademark of Motorola, Inc.

SCHEMATIC



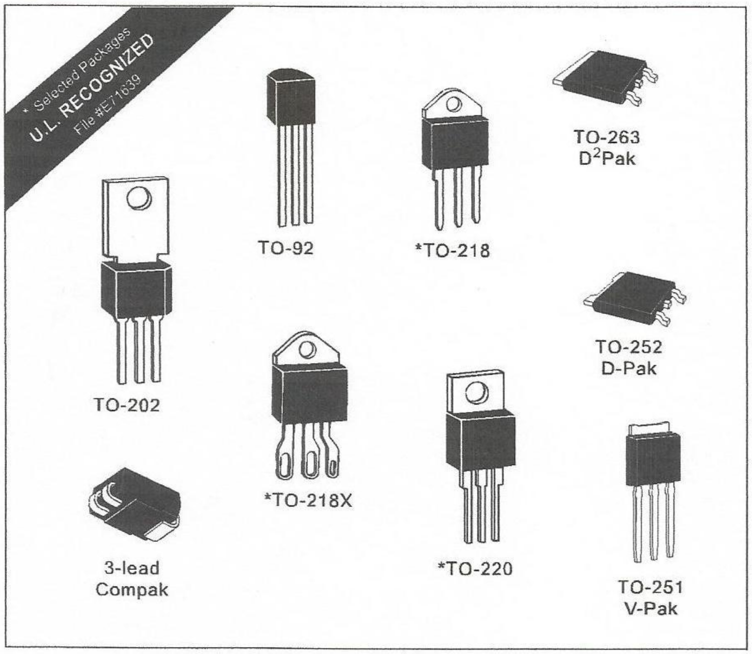
PIN 1. INPUT LED
2. INPUT LED
3. NO CONNECTION
4. EMITTER
5. COLLECTOR
6. BASE

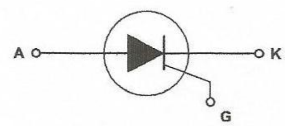
REV 1

MOTOROLA

Figure 60: LM7805 Data Sheet

E6





SCRs

(1 A to 70 A) RoHS

General Description

The Teccor line of thyristor SCR semi-conductors are half-wave, unidirectional, gate-controlled rectifiers which complement Teccor's line of sensitive SCRs. Teccor offers devices with ratings of 1 A to 70 A and 200 V to 1000 V, with gate sensitivities from 10 mA to 50 mA. If gate currents in the 12 µA to 500 µA ranges are required, see "Sensitive SCRs" section of this catalog.

Three packages are offered in electrically isolated construction where the case or tab is internally isolated to allow the use of low-cost assembly and convenient packaging techniques.

The Teccor line of SCRs features glass-passivated junctions to ensure long-term reliability and parameter stability. Teccor's glass offers a rugged, reliable barrier against junction contamination.

Variations of devices covered in this data sheet are available for custom design applications. Consult the factory for more information.

Features

- **RoHS Compliant**
- **Electrically-isolated package**
- **High voltage capability — 200 V to 1000 V**
- **High surge capability — up to 950 A**
- **Glass-passivated chip**

Compak SCR

- **Surface mount package — 1 A series**
- **New small profile three-leaded Compak package**
- **Packaged in embossed carrier tape with 2,500 devices per reel**
- **Can replace SOT-223**

Thyristor Product Catalog
E6 - 1
http://www.littelfuse.com
+1 972-580-7777

Figure 61: S6008R Data Sheet

1N4001 THRU 1N4007

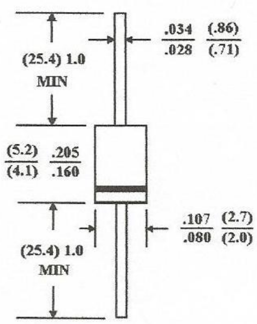
PLASTIC SILICON RECTIFIER

VOLTAGE - 50 to 1000 Volts CURRENT - 1.0 Ampere

FEATURES

- Low forward voltage drop
- High current capability
- High reliability
- High surge current capability
- Exceeds environmental standards of MIL-S-19500/228

DO-41



Dimensions in inches and (millimeters)

MECHANICAL DATA

Case: Molded plastic , DO-41
Epoxy: UL 94V-O rate flame retardant
Lead: Axial leads, solderable per MIL-STD-202,
method 208 guaranteed
Polarity: Color band denotes cathode end
Mounting Position: Any
Weight: 0.012 ounce, 0.3 gram

MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS

Ratings at 25 ¢J ambient temperature unless otherwise specified.
Single phase, half wave, 60 Hz, resistive or inductive load.
For capacitive load, derate current by 20%.

	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	UNITS
Maximum Recurrent Peak Reverse Voltage	50	100	200	400	600	800	1000	V
Maximum RMS Voltage	35	75	140	280	420	560	700	V
Maximum DC Blocking Voltage	50	100	200	400	600	800	1000	V
Maximum Average Forward Rectified Current .375"(9.5mm) Lead Length at T_A=75 ¢J				1.0				A
Peak Forward Surge Current 8.3ms single half sine-wave superimposed on rated load (JEDEC method)				30				A
Maximum Forward Voltage at 1.0A DC and 25 ¢J				1.1				V
Maximum Full Load Reverse Current Full Cycle Average at 75 ¢J Ambient				30				£g A
Maximum Reverse Current at T_A=25 ¢J At Rated DC Blocking Voltage T_A=100 ¢J				5.0 500				£g A £g A
Typical Junction capacitance (Note 1)				15				pF
Typical Thermal Resistance (Note 2) R £KJA				50				¢J/W
Typical Thermal resistance (NOTE 2) R £K JL				25				¢J/W
Operating and Storage Temperature Range T_J,T_{STG}				-55 to +150				¢J

NOTES:

1. Measured at 1 MHz and applied reverse voltage of 4.0 VDC.
2. Thermal Resistance Junction to Ambient and from junction to lead at 0.375"(9.5mm) lead length P.C.B mounted.

PANJIT

Figure 62: 1N4004 Data Sheet

Dimmer Software

```
//Authors: Qingyu Meng (Jace) and Stephanie Erickson
//Subroutine Declarations
#include    <pic18.h>
#include      "lcd_portd.h"
#include      "function.h"
#include      "function.c"
#include      "LCD_PortD.C"

unsigned int count = 0;
int PWM = 0;

const unsigned char MSG1[10] = "PWM =     ";

void LCD_Out(unsigned int DATA){  //write doubles to LCD
        unsigned char A[5], i;
        for (i=0; i<5; i++) {
                A[i] = DATA % 10;
                DATA = DATA / 10;
                }
        LCD_Write(ascii(A[4]));
        LCD_Write(ascii(A[3]));
        LCD_Write(ascii(A[2]));
        LCD_Write('.');
        LCD_Write(ascii(A[1]));
        LCD_Write(ascii(A[0]));
        }

unsigned char Keypad(void)  //controls power to keypad, reads data
{
        unsigned char RESULT;
        RESULT = 0xFF;
        TRISB = 0x78;
        PORTB = 4; // scan column #1
        if (RB6) RESULT = 1;
        if (RB5) RESULT = 4;
        if (RB4) RESULT = 7;
        if (RB3) RESULT = 10;
        PORTB = 2; // scan column #2
        if (RB6) RESULT = 2;
        if (RB5) RESULT = 5;
        if (RB4) RESULT = 8;
        if (RB3) RESULT = 0;
        PORTB = 1; // scan column #3
        if (RB6) RESULT = 3;
        if (RB5) RESULT = 6;
        if (RB4) RESULT = 9;
        if (RB3) RESULT = 11;
        return(RESULT);
}

unsigned int GetNumber(void)  //controls read from keypad
{
        unsigned int RESULT;
        unsigned int KEY;
        RESULT = 0;
        KEY = 0;
        while(KEY != 10) {
                while(Keypad() == 0xFF);
                KEY = Keypad();
                while(Keypad() != 0xFF);
                if (KEY < 10) RESULT = (RESULT * 10) + KEY;
        }
```

```
                return(RESULT);
}

void interrupt timer2(void){
                count++;
                count = count%100;
                if(count < PWM) RC0 = 1;        //bit bang to get PWM based on 1-100 resolution
                if(count >= PWM) RC0 = 0;
                TMR2IF = 0;  //reset flag
                }

// Main Routine
void main(void){
                unsigned char i;
                TRISA = 0;
                TRISB = 0xFF;
                TRISC = 0;
                TRISD = 0;
                ADCON1 = 0x0F;
                LCD_Init();

                // initialize Timer2
                T2CON = 0x05;
                PR2 = 207;
                TMR2IE = 1;
                PEIE = 1;
                TMR2ON = 1;
                TMR2IP = 1;
                // Turn on all interrupts
                GIE = 1;
   LCD_Move(0,0); for(i=0; i<10; i++) LCD_Write(MSG1[i]);

while(1)
{
   if(RB0&&PWM>=5)
                PWM=PWM-5;
                 LCD_Move(1,0); LCD_Out(PWM);
   if(RB1&&PWM<=95)
                PWM=PWM+5;
                LCD_Move(1,0); LCD_Out(PWM);
   if(RB2)
   {
                while(PWM>0)
   {
  PWM--;
  LCD_Move(1,0); LCD_Out(PWM);
  Wait_ms(100);
   }
   }
  if(RB3)
   {
                 while(PWM<100)
   {
  PWM++;
  LCD_Move(1,0); LCD_Out(PWM);
  Wait_ms(100);
   }
   }

 if(RB4)
   {
                while(PWM>0)
   {
```

```
  PWM--;
  LCD_Move(1,0); LCD_Out(PWM);
  Wait_ms(1000);
   }
   }
 if(RB5)
   {
            while(PWM<100)
   {
  PWM++;
  LCD_Move(1,0); LCD_Out(PWM);
  Wait_ms(1000);
   }
   }

}
}
```